Object detection and segmentation on NICOL robot

Laura Jouvet

University supervisor : Jan-Gerrit Habekost

Academic supervisor : Pr. Luc Jaulin

Second year internship – May-August 2024

Table of Contents

Acknowledgments **2**

Abstract **3**

1 Introduction **4**

1.1 The WTM research group 4

1.2 NICO and NICOL robots 5

1.3 My project 6

2 Proposal for the research topic **7**

2.1 Related Work 7

2.2 Approach 9

2.3 Aim of the research 10

3 NICOL environment **11**

3.1 Architecture 11

3.2 Simulation 12

3.3 Tests codes 12

4 3D part **13**

4.1 Integration of three cameras 13

4.2 Calibration of the cameras 14

5 Dataset **17**

5.1 Dataset collection 17

5.2 Labels 18

6 Comparison of objects detectors **19**

6.1 The IoU method 19

6.2 The chosen parameters 19

6.3 The results 21

7 Add of SAM **24**

7.1 SAM after detection 24

7.2 SAM before detection 25

8 Conclusion **27**

8.1 Conclusion on the research topic 27

8.2 Conclusion on the internship 27

References **29**

Acknowledgments

I would like to thank the Knowledge Technology research group of the Hamburg University in Germany for providing me with the opportunity to undertake my internship with them.

First, special thanks to my supervisor, Jan-Gerrit Habekost, who has always made time for me or found someone to guide me if he could not be here. Thanks to him, I could work on various tasks and learn a lot during my internship.

Thanks also to Pr. Wermter, who always paid attention to integrate me well in the group and in the research activities.

Finally, thanks to the other members of the WTM group who spent some time with me to explain me their work and to answer my questions. Thanks to them and to the activities that they organized outside of work, I have always felt at ease in the group, which allowed me to integrate well and to learn a lot on different subjects.

Abstract

This report presents my four-month internship in the Knowledge Technology research group of the Hamburg University in Germany.

The aim of this internship was to find a research topic in the field of object detection with NICOL robot and to work on this subject. The internship involved algorithms development and testing in a simulated environment using ROS and Gazebo, integration of 3D cameras on NICOL robot and research about objects detectors.

This multidisciplinary internship encompassed simulation, hardware integration, project management and collaboration within a research group. Finally, it contributed to improve my English language proficiency and presentation skills, and gave me a great cross-cultural experience.

Résumé

Ce rapport présente mon stage de quatre mois au sein du groupe de recherche Knowledge Technology de l'université de Hambourg en Allemagne.

Le but de ce stage était de trouver un sujet de recherche dans le domaine de la détection d'objets avec le robot NICOL et de travailler sur ce sujet. Le stage a impliqué le développement et le test d'algorithmes dans un environnement simulé à l'aide de ROS et de Gazebo, l'implémentation de caméras 3D sur le robot NICOL et de la recherche sur les détecteurs d'objets.

Ce stage multidisciplinaire a englobé de la simulation, de l'intégration matérielle, de la gestion de projet et de la collaboration au sein d'un groupe de recherche. Enfin, il a contribué à l'amélioration de ma maîtrise de la langue anglaise, de mes compétences en présentation et cela m'a fourni une excellente expérience interculturelle.

1 Introduction

1.1 The WTM research group

I made my internship in the Knowledge Technology (WTM) research group of the University of Hamburg in Germany. This group is a part of the 9 research groups working on the field of Human-Centered Computing (HCC) in this university.

The objective of the WTM group is to research into artificial intelligence and hybrid knowledge technology based on learning neural networks and explainable representations to build novel intelligent systems and robot assistants. Such systems include for instance adaptive interactive knowledge discovery systems, learning crossmodal neural agents with vision and language capabilities, or neuroscience-inspired continually learning robots. Their approach is often motivated by nature and especially from the brain, cognition and neuroscience.

The WTM group is well known across the world. Its members often go abroad to present their papers during some conferences and searchers from a lot of foreign universities come regularly in Hamburg to work with the group. Recently, the WTM group had a paper accepted at the 33rd International Conference on Artificial Neural Networks (ICANN), an annual conference organized by the European Neural Network Society (ENNS), which took place at the end of September.

Figure 1 – WTM building

Throughout my four-month internship, I was able to talk with a lot of people in this group and it allowed me to discover the variety of its projects that I will detail in the next subsection. Moreover, thanks to Pr. Wermter, the director of the research group and to Jan-Gerrit Habekost, my supervisor, who really wanted me to discover all the aspects of a research group, I could participate into a lot of group activities like group meetings, lectures or Master and PHD defences. Finally, one of the things that I really liked too was that people in this group were coming from a lot of different countries so it was very interesting to see how they can communicate and work with each other and bring together their knowledge and different ways of thinking in the service of research and science.

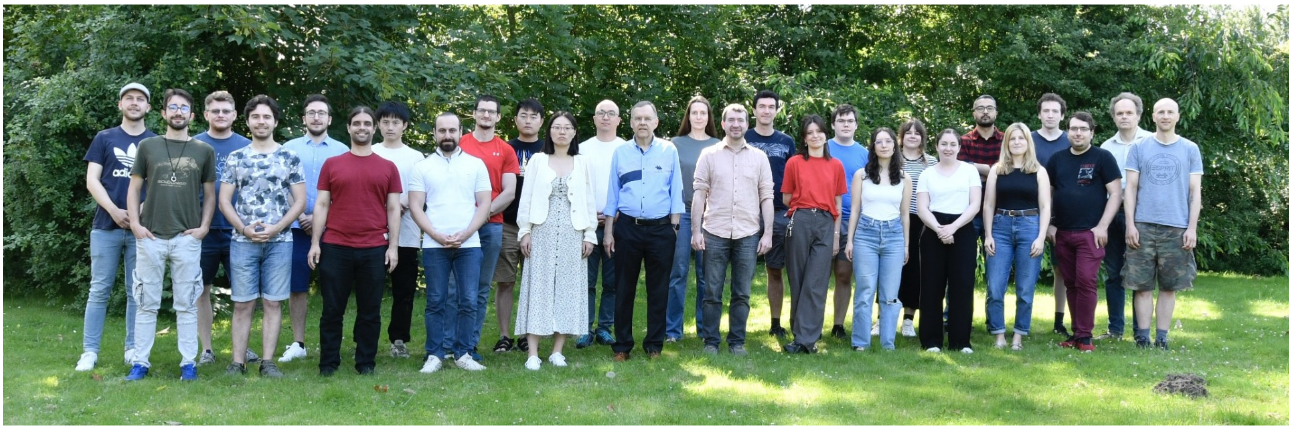


Figure 2 – WTM members

1.2 NICO and NICOL robots

Most of WTM group members work on NICO and NICOL robots. NICO stands for Neuro-Inspired COmpanion. This is a humanoid developmental robot and a neuro-cognitive research platform for embodied sensorimotor computational and cognitive models in the context of multimodal interaction. It fills a gap between necessary sensing and interaction capabilities and flexible design. Research is developed on NICO in many directions such as objects grasping, objects detection and some NICO robots have even legs to train to walk.

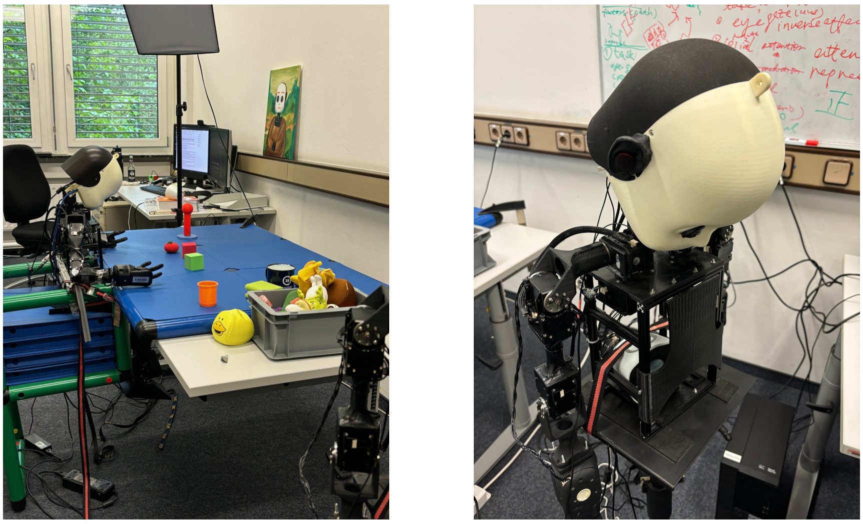


Figure 3 – NICO robot

However, one of the most important problem on NICO was that most of objects were to heavy for its arms. This is why the group then created NICOL robot which stands for Neuro-inspired Collaborative Semi-humanoid Robot. This robot is taller than NICO and has only a bust fixed to a structure as well as two large arms. It adopts NICO's head and facial expression display, and extends its manipulation abilities in terms of precision, object size and workspace size. The aim of NICOL is to bridge social interaction and reliable manipulation Indeed, many existing robotic platforms are either designed for social interaction or industrial object manipulation tasks while the design of NICOL and collaborative robots allow them to both emphasize their social interaction and their physical collaboration abilities.

In the WTM group, searchers are working with NICOL on objects grasping, objects detection, communication with humans and imitation of the movements of a interlocutor. To do that, they developed and extended different neural and hybrid neuro-genetic visuomotor approaches initially developed for the NICO robot. Then, they presented a novel neuro-genetic approach that improves the grasp-accuracy of NICOL to over 99 percent.

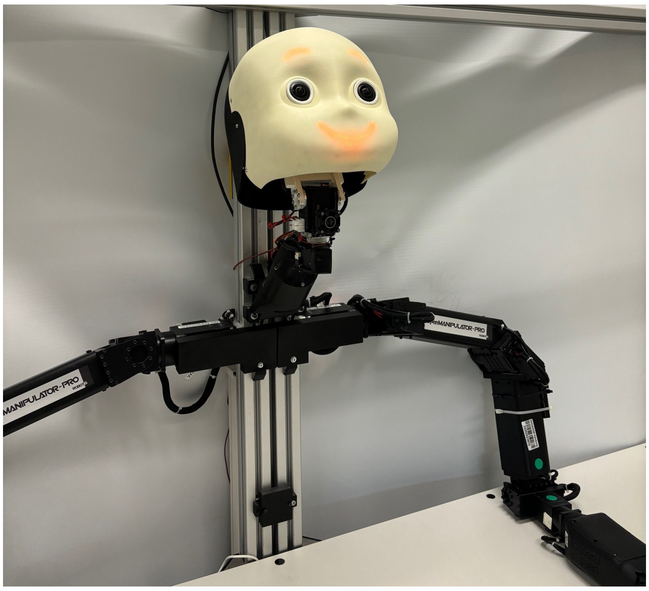


Figure 4 – NICOL robot

One of the next objectives of the group is to adapt the codes written on NICOL so that they can be used on both robots without any change.

1.3 My project

During my internship, Jan-Gerrit Habekost offered me to work on NICOL robot and more particularly on object detection. Pr. Wermter and him wanted me to participate in the group activities like the other searchers so I first had to write a proposal and to give a presentation of my research topic in front of the research group and some other members of the university.

Then, during the first weeks of my internship, I had to define more precisely the subject I will work on and to write the proposal. I made some research about the state of the art and then we decided to create an architecture for my research project. In the rest of this report, you will see the initial proposal and how we have developed it during the internship. At the end of the internship I also gave a presentation in front of the WTM team to present the progress of my research topic.

2 Proposal for the research topic

Object detection has become indispensable in many fields of computer vision like autonomous vehicles, security or augmented reality. One of the most widely used methods for object detection is image segmentation and, since April 2023, a model which uses this method has gained popularity. This model created by Meta is called the Segment-Anything Model (SAM) [5]. Its two main highlights are its impressive capabilities in various segmentation tasks and its prompt-based interface.

A lot of research is being done around SAM, particularly in the medical field [6], because for the moment the model lacks training in this area, where it could be very useful. In this project, we want to apply this research to NICOL robot on object detection with objects of the daily life. Indeed, we thought that SAM could bring a lot to NICOL on this task.

2.1 Related Work

SAM is a promptable segmentation system with zero-shot generalization to unfamiliar objects and images, without the need for additional training. It was built on the largest segmentation dataset so far, comprising over 1 billion ground-truth segmentation masks on 11 million licensed and privacy-respecting natural images [3].

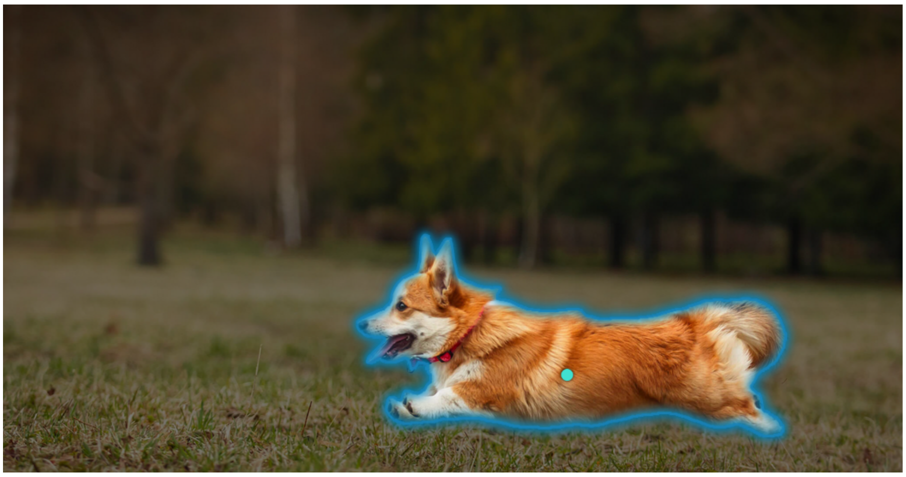


FIGURE 5 – Segmentation of a dog with SAM [5]

2.1.1 The SAM architecture

Regarding the SAM architecture, a powerful image encoder computes an image embedding and a prompt encoder embeds prompts. The prompt encoder can be either sparse (points, boxes) or dense (masks). Then, the two information sources are combined in a lightweight mask decoder that predicts segmentation masks.

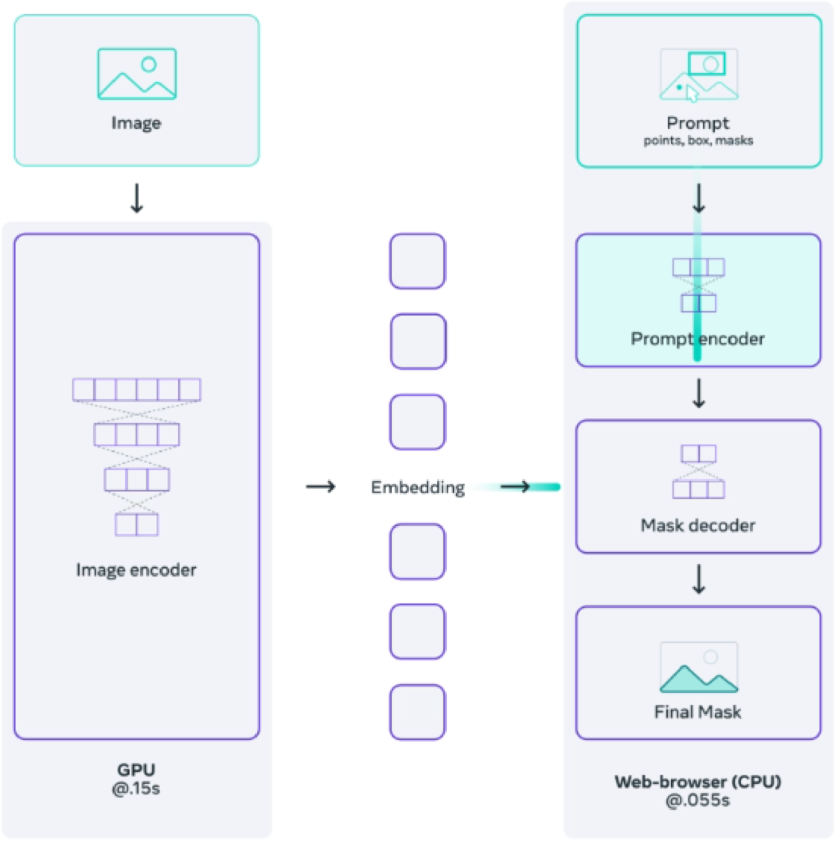


FIGURE 6 – The SAM architecture [5]

2.1.2 Text prompts

SAM would also be able to process text prompts. Indeed, according to Meta, the user could use the CLIP text encode without any modification to create a prompt for SAM. However, if Meta explored text prompts in its work, its capabilities have not been released yet [3].

2.2 Approach

To use SAM on a 2D image, there are different prompts possibilities : points, boxes or masks. For instance, SAM can be directly applied on the image and it will find a lot of masks to detect each object of the picture, or a point can be chosen on the image and it will find a single mask to detect this specific object. Obviously, to detect objects, it is more precise to choose a point, than to apply SAM to the whole image [3].

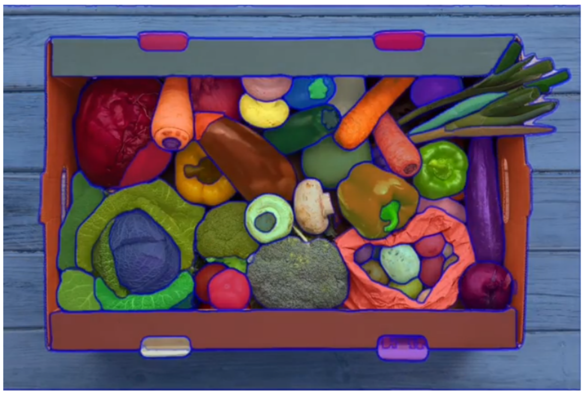


FIGURE 7 – SAM with masks [5]

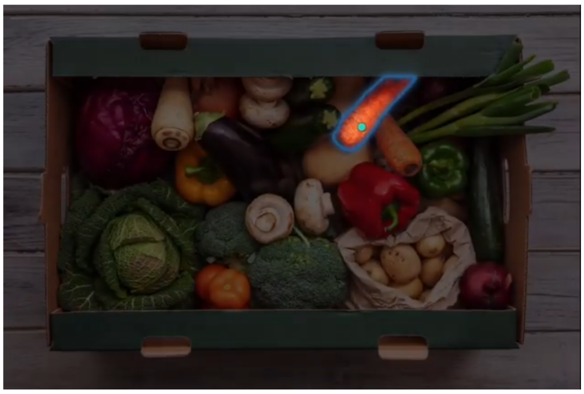


FIGURE 8 – SAM with points [5]

To apply SAM to NICOL robot, we could just use SAM with RGB cameras because it already does a good job in detecting objects but, in this project, we want to try to improve the results by adding candidates points, as it could be done manually on a 2D image.

First, we will get pointclouds from a depth estimator model used by the WTM group or directly from the Realsense. Then, we will remove all points that are not really elevated from the table by using a treshold of 2 cm and we will obtain clusters of points so we will have an estimation of the 3D objects location.

Then, we will need to localize the pixels in the 2D eye image that correspond to these 3D locations. To do that, we will use the camera model written by a member of the WTM group and we will obtain candidates points. These points will play the role of the points that could have been manually selected on a 2D image.

Thus, we will use these candidate points with SAM : we will enter a candidate point and get a single mask around it. It will enable us to find the class of the objects that we would have detected.

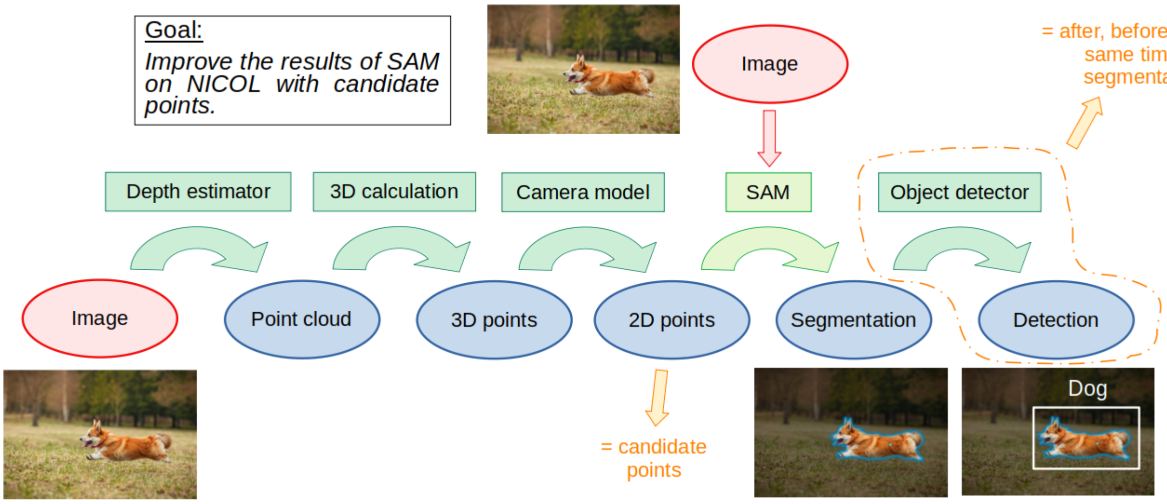


FIGURE 9 – Architecture of the approach

The architecture of this approach is innovative in the field of object detection because most of research in this field is done on 2D images. However, in our human environment we always see things in 3D. This is why Jan-Gerrit Habekost and Pr. Wermter wanted to start to work in 3D, to be closer to human perception.

2.3 Aim of the research

Using the approach described above, we will be able to determine whether the candidate points have improved the results of SAM on NICOL or not. If it is the case, it will mean that this method using SAM is very efficient to detect objects with NICOL robot.

The next step could be to work on text prompts. Indeed, even if its capabilities have not been released yet by Meta, text prompts should work on SAM [3]. In the case of the NICOL robot, it could be very interesting and useful to have such prompts to detect objects with open-vocabularies.

Then, it could also be interesting to search if it is possible to use SAM directly on 3D images with NICOL. In his paper, Wu et al. propose Space-Depth Transpose (SD-Trans) to adapt 2D SAM to 3D medical images and Hyper-Prompting Adapter (HyP-Adpt) to achieve prompt-conditioned adaptation [6]. If this method could be apply to SAM, it would avoid the need to transpose 3D objects locations to 2D image pixels.

3 NICOL environment

The NICOL robot was built by some researchers of the WTM group and all the codes are opensource. It has an API and a ROS and Coppelia parts and we can simulate the platform and the robot in Gazebo.

3.1 Architecture

There are three parts in the NICOL architecture : the NICOL API, the ROS part and the Coppelia part. During my internship I only used the two first ones.

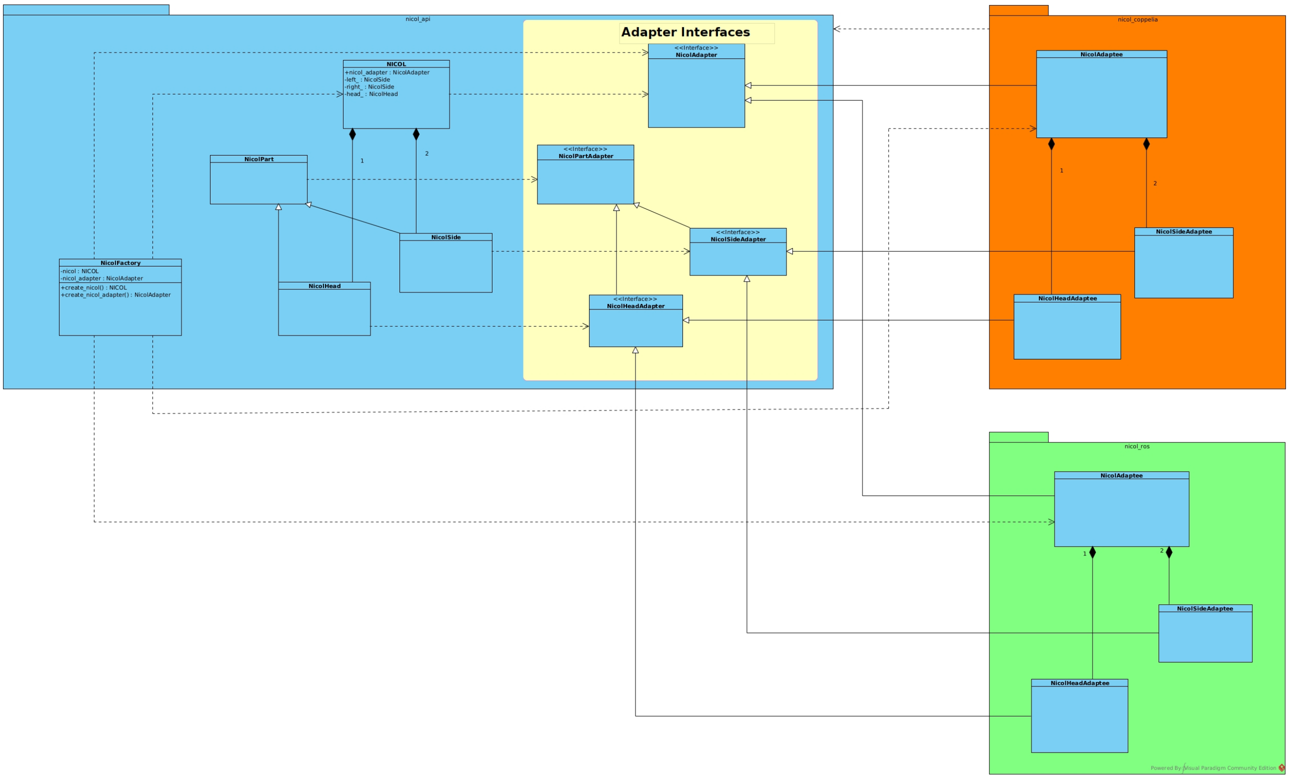


FIGURE 10 – NICOL architecture

In the NICOL API in blue, there are three parts in the code : the NicolSide with the settings of the arms and the hands, the NicolHead with the settings of the head and the NicolPart where these to previous parts are gathered. The adapter interfaces in yellow allows to adapt the code in another environment much more quickly (it takes around a week with the adapter interfaces and around 6 months without it). The NICOL API was coded in a year and then the group had to adapt it to ROS and Coppelia. In green there is the ROS part in which we find again a part for NICOL's sides and a part for its head and this is the same thing for Coppelia in orange.

3.2 Simulation

To use NICOL, there is also a Gazebo simulator in which we can see the platform and create and move some objects. For instance on the left image we put an object on the table in front of NICOL and on the right image, we can see the pointcloud of this object.

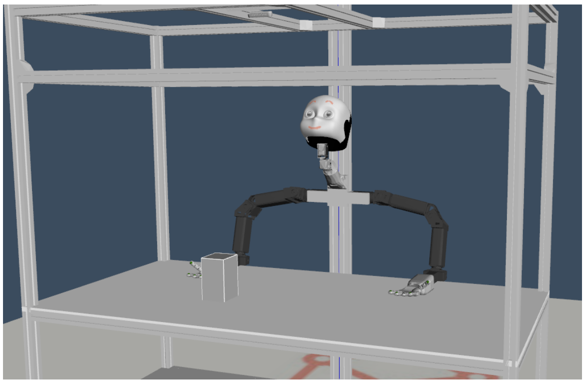


FIGURE 11 – NICOL simulator

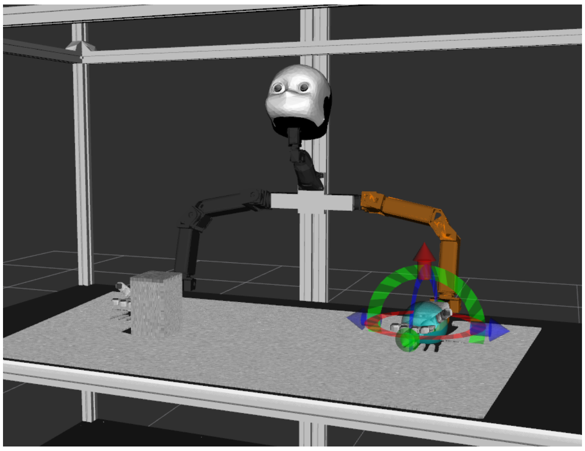


FIGURE 12 – Pointcloud in the simulator

In our research topic, we wanted to have this pointcloud with the real NICOL and for that we needed to install 3D cameras which is explained in the part 4 of this report.

3.3 Tests codes

To understand better how NICOL API and the ROS part work, we worked a little bit on NICOL tests codes and we wrote codes to test NICOL's sides and NICOL's head.

The principle of these tests is for instance to give a goal position for one arm of NICOL so we give numbers for the 8 joints of the arm. Then, NICOL will take a position and thanks to a function called *inverse_kinematics* we can recover the joints values from these position and orientation. To validate the test, the difference between the initial values of the joints and the reached ones must be small.

These codes are very useful because it allows to test a lot of positions on the real NICOL robot and to be sure that the given positions are reached by NICOL. We tested them in the Gazebo simulation and it worked.

4 3D part

The work about 3D vision was new in the WTM group so we had to add three RGBD cameras on NICOL's set because until now there were just two RGB fisheye cameras on the head of NICOL and one RGBD camera above the table.

4.1 Integration of three cameras

We decided to add one RGBD camera on NICOL's head thanks to a 3D printed head band and two other RGBD cameras on the right and on the left of the table. Indeed, it was not enough to have only one RGBD camera above the table and one on the head of NICOL because it did not allow us to have the whole objects. As we can see it in the simulation, we would have had holes in the pointcloud. Then, we added also two RGBD cameras on both sides of the table.

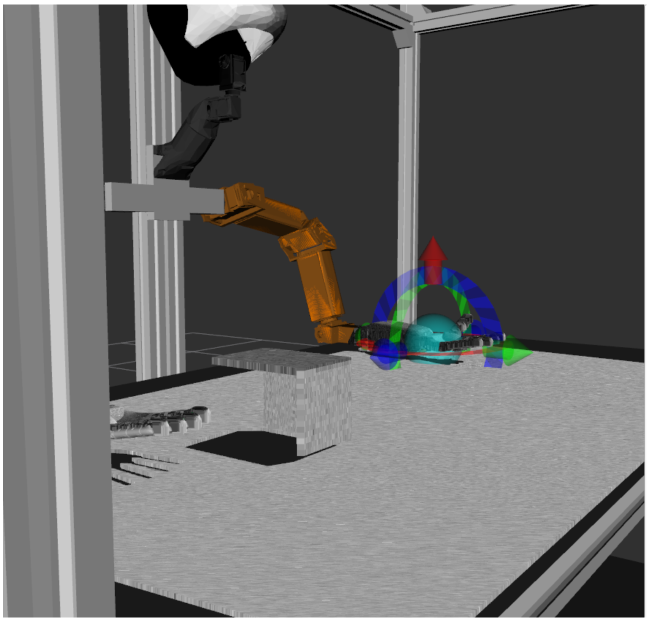


Figure 13 – Hole in the pointcloud

First, we had to implement the three new cameras into the NICOL API thanks to ROS topics and to create tfs to move and orientate them. Each time, we had to separate RGB and deep parts. Finally, we launched the code on the real NICOL and we succeeded in obtaining a visualization of the pointcloud with the four cameras.

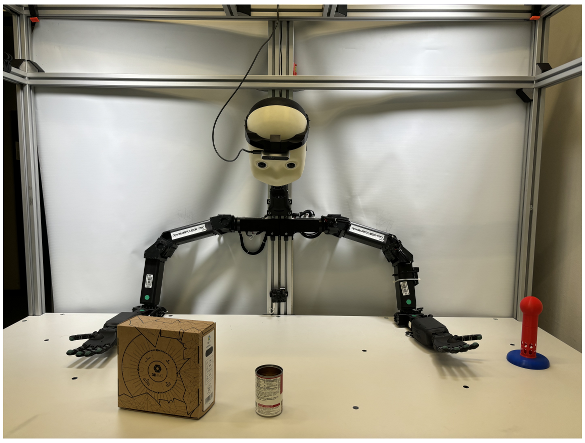


Figure 14 – Cardboard and can

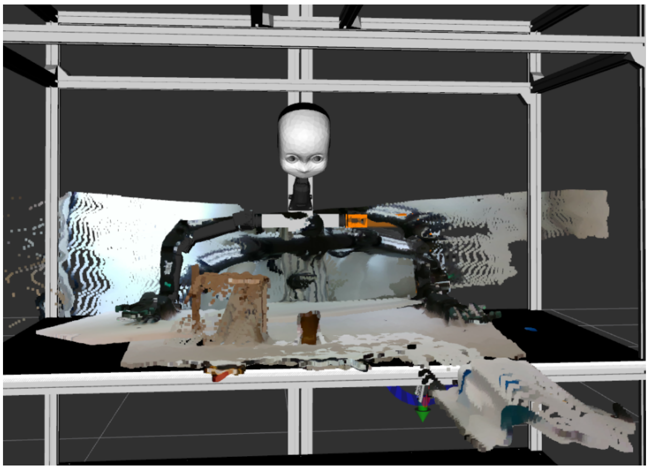


Figure 15 – Real pointcloud

4.2 Calibration of the cameras

Then, we had to calibrate the cameras to find their real positions in order to have aligned plans in the pointcloud. Indeed, in the figure 15 we can see that some plans are not aligned with the table and this is why the cardboard on the table is not perfect.

To calibrate the camera on the head, we had a code bases on reprojection error. We had 29 points on the table with known coordinates and the principle is that we take a point whose position is known, we project it into the 3D space and then we reproject it on the table. At the end, the aim is to have the lowest difference between this final position and the real position of the point.

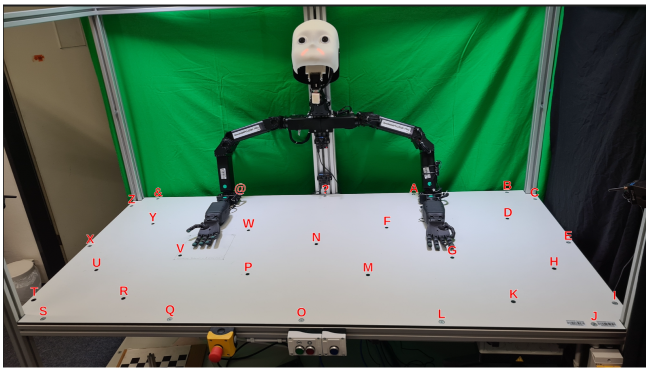


Figure 16 – Points for the calibration

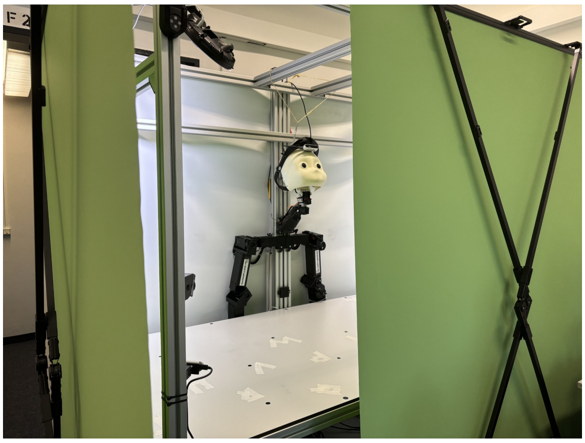


Figure 17 – Calibration setup

During the calibration, we first have to save a video of the camera while the head of NICOL is moving in order to have a lot of points on the screen. To do that, we created a rosbag. Then, in the code there is a detection of the points and when a point is detected, a number appears on the screen (figure 18). To correctly detect the points, we had to choose some parameters in order to have the good light to see the points on the table. We also needed to only detect the points and no parasitic objects. To answer these two constraints, we placed some green panels around NICOL as we can see in the figure 17.

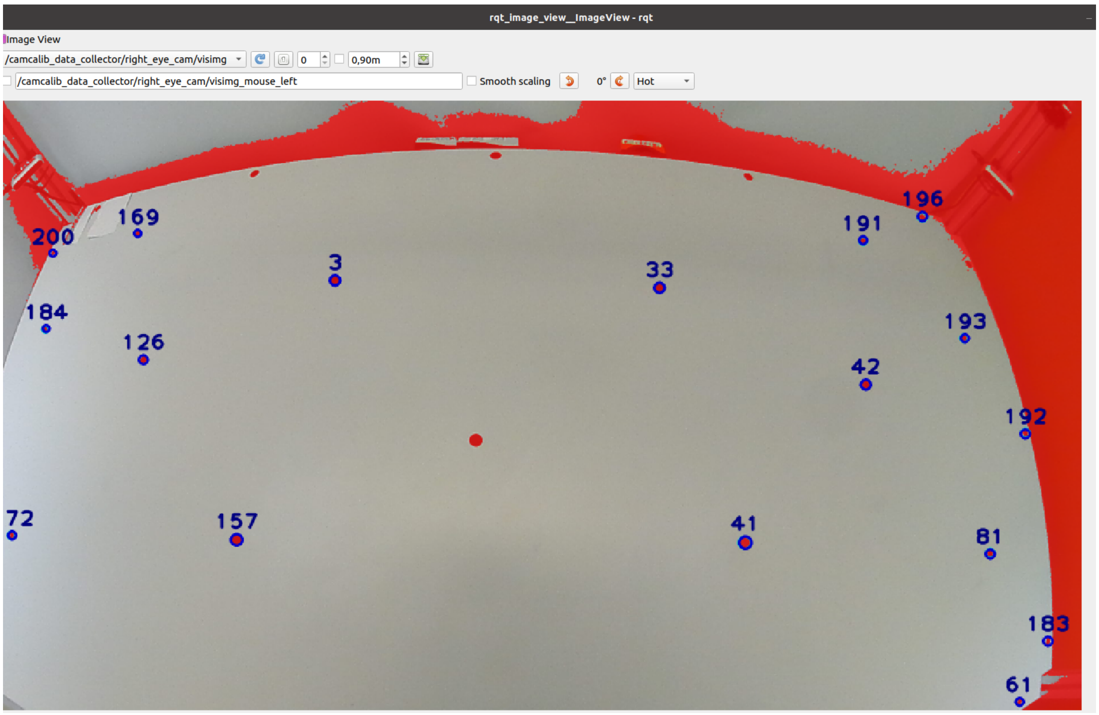
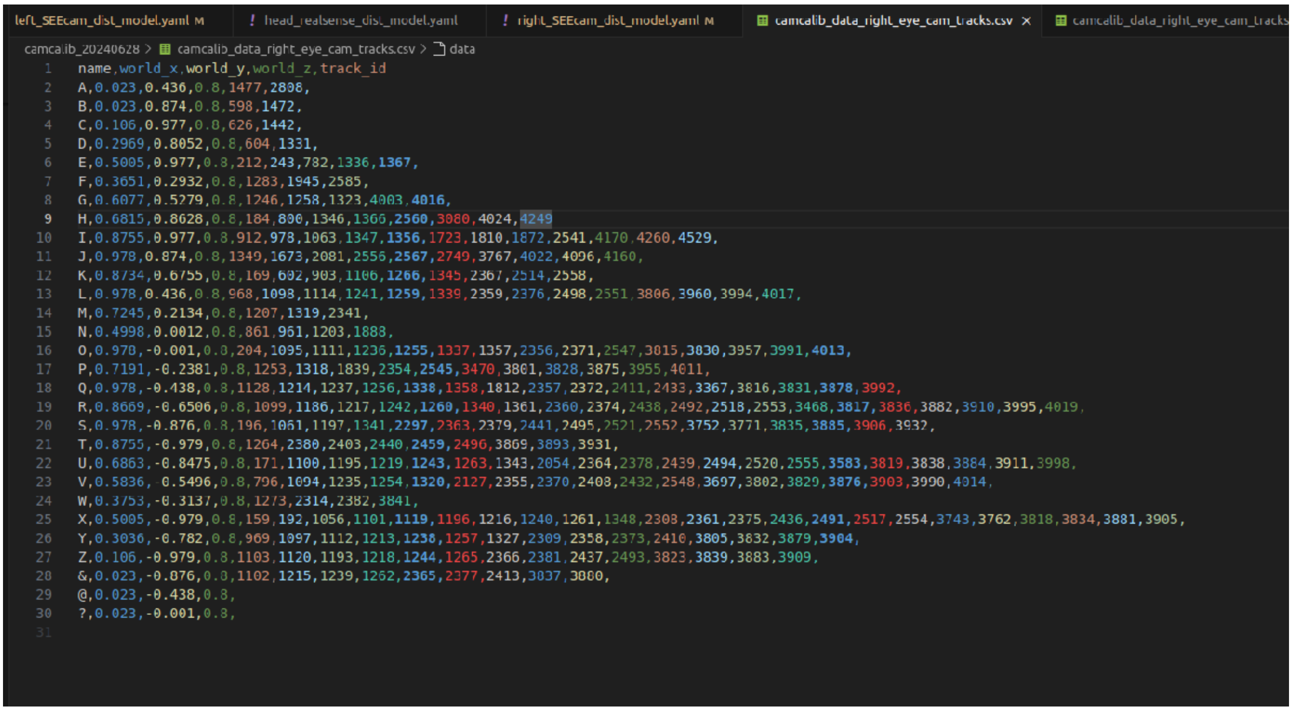

rqt_image_view__ImageView - rqt
Image View
/camcalib_data_collector/right_eye_cam/visimg
0
0,90m
/camcalib_data_collector/right_eye_cam/visimg_mouse_left
Smooth scaling
0°
Hot
200
169
3
33
191
196
184
126
193
42
192
72
157
41
81
183
61

FIGURE 18 – Detected points

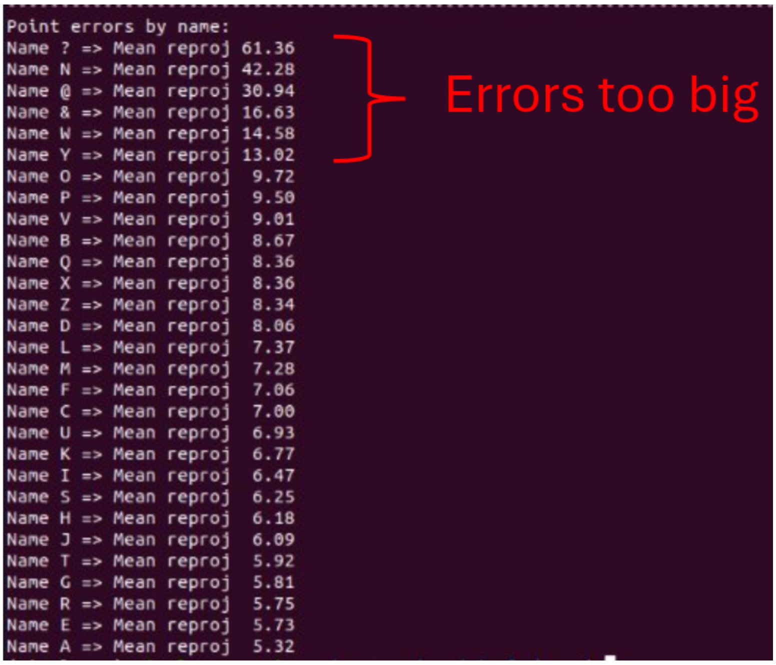
Then, we had to play the rosbag and for each detection we put the written number in the column of the letter corresponding to the detected point in a file. To do this we always had to be two at the same time behind the computer : one person who wrote the number in the good column and another one who checked if the other had written the right things at the right places. Indeed, this step is very long but it is very important to be focused and careful because if one point is wrong, the calibration will be too bad.


left_SEEcam_dist_model.yaml M
head_realsense_dist_model.yaml
right_SEEcam_dist_model.yaml M
camcalib_data_right_eye_cam_tracks.csv
camcalib_data_right_eye_cam_tracks
camcalib_20240628 > camcalib_data_right_eye_cam_tracks.csv > data

```
name,world_x,world_y,world_z,track_id
A,0.023,0.436,0.8,1477,2808,
B,0.023,0.874,0.8,598,1472,
C,0.106,0.977,0.8,626,1442,
D,0.2969,0.8052,0.8,604,1331,
E,0.5005,0.977,0.8,212,243,782,1336,1367,
F,0.3651,0.2932,0.8,1283,1945,2585,
G,0.6077,0.5279,0.8,1246,1258,1323,4003,4016,
H,0.6815,0.8628,0.8,184,800,1346,1366,2560,3080,4024,4249
I,0.8755,0.977,0.8,912,978,1063,1347,1356,1723,1810,1872,2541,4170,4260,4529,
J,0.978,0.874,0.8,1349,1673,2081,2556,2567,2749,3767,4022,4096,4160,
K,0.8734,0.6755,0.8,169,602,903,1106,1266,1345,2367,2514,2558,
L,0.978,0.436,0.8,968,1098,1114,1241,1259,1339,2359,2376,2498,2551,3806,3960,3994,4017,
M,0.7245,0.2134,0.8,1207,1319,2341,
N,0.4998,0.0012,0.8,861,961,1203,1888,
O,0.978,-0.001,0.8,204,1095,1111,1236,1255,1337,1357,2356,2371,2547,3815,3830,3957,3991,4013,
P,0.7191,-0.2381,0.8,1253,1318,1839,2354,2545,3470,3801,3828,3875,3955,4011,
Q,0.978,-0.438,0.8,1128,1214,1237,1256,1338,1358,1812,2357,2372,2411,2433,3367,3816,3831,3878,3992,
R,0.8669,-0.6506,0.8,1099,1186,1217,1242,1260,1340,1361,2360,2374,2438,2492,2518,2553,3468,3817,3836,3882,3910,3995,4019,
S,0.978,-0.876,0.8,196,1061,1197,1341,2297,2363,2379,2441,2495,2521,2552,3752,3771,3835,3885,3906,3932,
T,0.8755,-0.979,0.8,1264,2380,2403,2440,2459,2496,3869,3893,3931,
U,0.6863,-0.8475,0.8,171,1100,1195,1219,1243,1263,1343,2054,2364,2378,2439,2494,2520,2555,3583,3819,3838,3884,3911,3998,
V,0.5836,-0.5496,0.8,796,1094,1235,1254,1320,2127,2355,2370,2408,2432,2548,3697,3802,3829,3876,3903,3990,4014,
W,0.3753,-0.3137,0.8,1273,2314,2382,3841,
X,0.5005,-0.979,0.8,159,192,1056,1101,1119,1196,1216,1240,1261,1348,2308,2361,2375,2436,2491,2517,2554,3743,3762,3818,3834,3881,3905,
Y,0.3036,-0.782,0.8,969,1097,1112,1213,1238,1257,1327,2309,2358,2373,2410,3805,3832,3879,3904,
Z,0.106,-0.979,0.8,1103,1120,1193,1218,1244,1265,2366,2381,2437,2493,3823,3839,3883,3909,
&,0.023,-0.876,0.8,1102,1215,1239,1262,2365,2377,2413,3837,3880,
@,0.023,-0.438,0.8,
?,0.023,-0.001,0.8,
```

FIGURE 19 – Numbers of the detected points

When we had finished, we ran a code that calculates the reprojection error. Unfortunately, the error was too big for 6 points. Indeed, on the screenshot we can see that the points ?, N, @, &, W, and Y have an error between 10 and 60. Even the other results between 5 and 10 are not really good because another member of the WTM group had already succeeded in having a result of 2. It was not with the same camera but for us the aim was to have an error between 3 and 5.



```
Point errors by name:
Name ? => Mean reproj 61.36
Name N => Mean reproj 42.28
Name @ => Mean reproj 30.94
Name & => Mean reproj 16.63
Name W => Mean reproj 14.58
Name Y => Mean reproj 13.02
Name O => Mean reproj  9.72
Name P => Mean reproj  9.50
Name V => Mean reproj  9.01
Name B => Mean reproj  8.67
Name Q => Mean reproj  8.36
Name X => Mean reproj  8.36
Name Z => Mean reproj  8.34
Name D => Mean reproj  8.06
Name L => Mean reproj  7.37
Name M => Mean reproj  7.28
Name F => Mean reproj  7.06
Name C => Mean reproj  7.00
Name U => Mean reproj  6.93
Name K => Mean reproj  6.77
Name I => Mean reproj  6.47
Name S => Mean reproj  6.25
Name H => Mean reproj  6.18
Name J => Mean reproj  6.09
Name T => Mean reproj  5.92
Name G => Mean reproj  5.81
Name R => Mean reproj  5.75
Name E => Mean reproj  5.73
Name A => Mean reproj  5.32
```

Figure 20 – Results of the reprojection error

Then, we started again to run the rosbag and we rewrote the numbers in the columns. This time we had just two points with huge errors so we removed these points from the columns and we had a good result of 3.88. Finally, the code gave us the matrix of the positions and orientations of the camera and we just had to write these parameters in our code.

After that, we had a problem with the two cameras on the left and on the right of the table because we realized that they were too low compared to the table. But the problem was that we had stuck them to 3D printed parts so we needed to print them again. We had no time for that so we decided to abandon these cameras for the internship.

5 Dataset

At this point of our work, we had to quicly collect a dataset for the next steps because someone needed to use NICOL without its head band but if we had removed it, we would have had to recalibrate it.

5.1 Dataset collection

To collect this dataset, we used the RGBD camera on the head of NICOL and the two RGB fisheye cameras. In order not to have problems with brightness or reflections on objects, we put back the large green panels around NICOL.

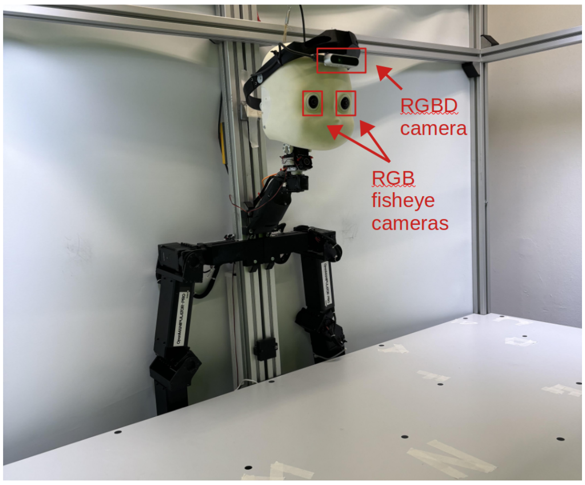


Figure 21 – Used cameras

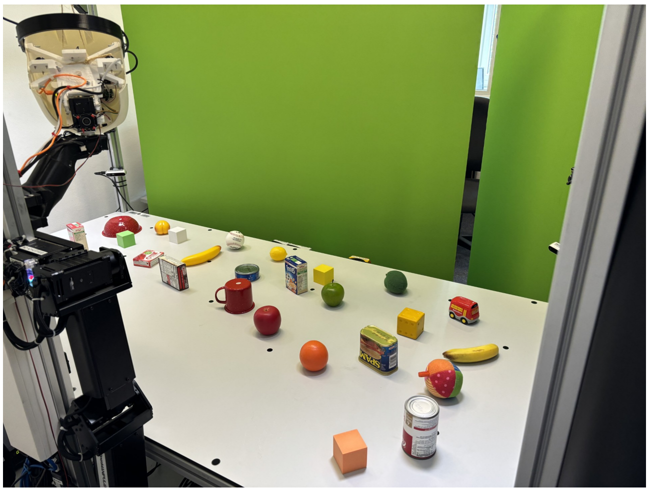


Figure 22 – Dataset collection setup

During a week, we collected 15 000 images divided into 300 scenes. For each scene, we moved the objects on the table and then we ran a code that makes move the head of NICOL from the right to the left to take 50 pictures. We created two kind of scenes (scenes with spaced objects and scenes with piled ones) in order to have easy scenes and more difficult ones for the rest of our work.

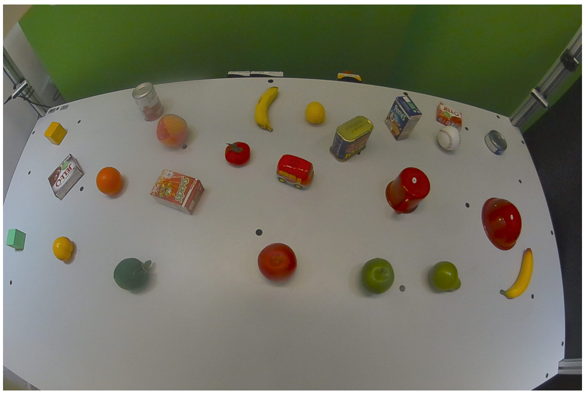


Figure 23 – Spaced objects

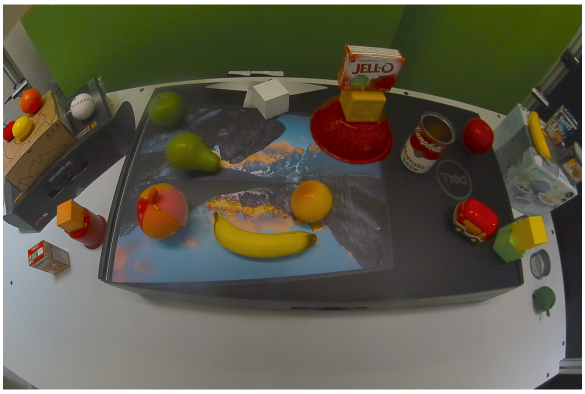


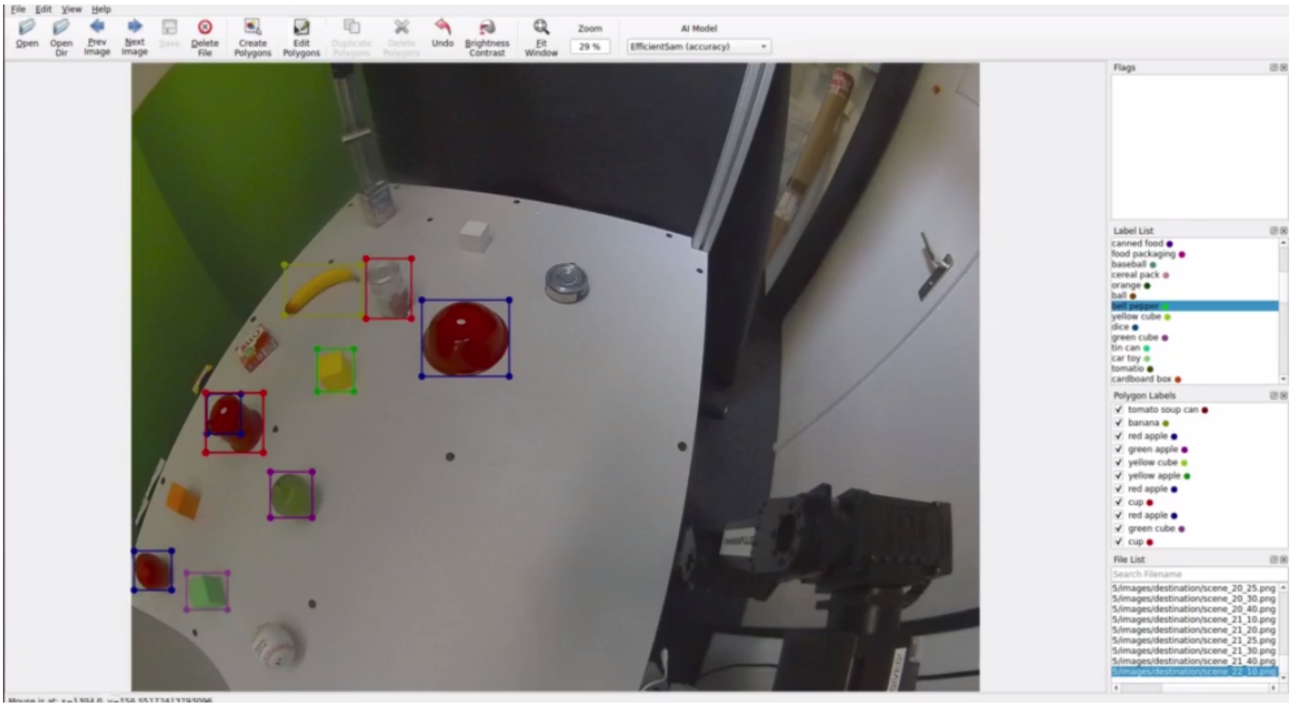
Figure 24 – Piled objects

5.2 Labels

When we had our dataset, we looked at our research architecture and a question came to our mind : before continuing to focus on the 3D part, which objects detector will be the best with NICOL and the objects that we had ? Then, in order to train and to test objects detectors with these images, we had to label some of them. It would also be a great thing for the group to have such a huge dataset and some labeled images.

To do this, we installed LabelMe and to compare objects detectors in the time left for my internship we decided that it would be good to label 1000 images. But it would have been too long to label them all by hand so we decided to first use an objects detector and chose YOLO-World to do that. Then, we entered the names of all the objects on YOLO-World and gave it our 1000 images. Finally, we wrote some codes to convert the output in a json file that could be read by Labelme.

The next step on LabelMe was to check if the detected objects were correctly detected and labeled and to label the objects that were not detected which was the case for a lot of them.



Zoom 29 %

AI Model EfficientSam (accuracy)

Figure 25 – Labels on Labelme

Finally, after two weeks we had 1000 labeled images.

6 Comparison of objects detectors

To find which objects detector was the best one with NICOL, we chose to compare the most famous ones : YOLO-World [1], OwlV2 [4] and the Vision and Language Knowledge Distillation (ViLD) [2].

6.1 The IoU method

To compare the three objects detectors, we used the Intersection over Union (IoU) method. In this method, we use two bounding boxes : the ground-truth bounding box which is the bounding box from the labeled image and the predicted bounding box which is the output of the objects detector.

Then, the IoU score is the surface of the area of overlap between the two bounding boxes divided by the surface of the union of the two bounding boxes.

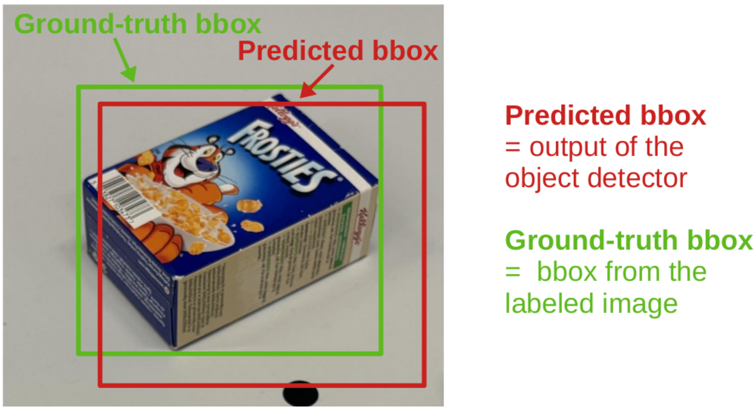


FIGURE 26 – Predicted and ground-truth bbox

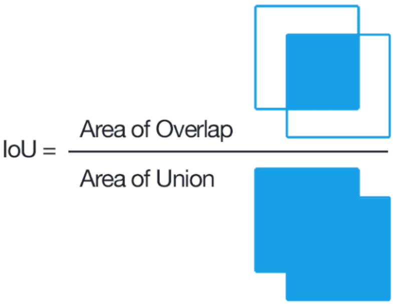


FIGURE 27 – The IoU method

The result is a number between 0 and 1. If the IoU score is close to 0, it means that the detection is really bad while if it is close to 1, it means that it is an excellent detection. An IoU score above 0.5 is already considered as a good detection.

6.2 The chosen parameters

To evaluate the objects detectors with the IoU method, we decided to use some specific parameters :

- Mean of true positives : objects which were correctly detected
- Mean of false positives : objects which were detected while they were not the objects on the table
- Mean of false negatives : objects which were not detected
- Mean of IoU scores
- Percentage of correctly detected objects

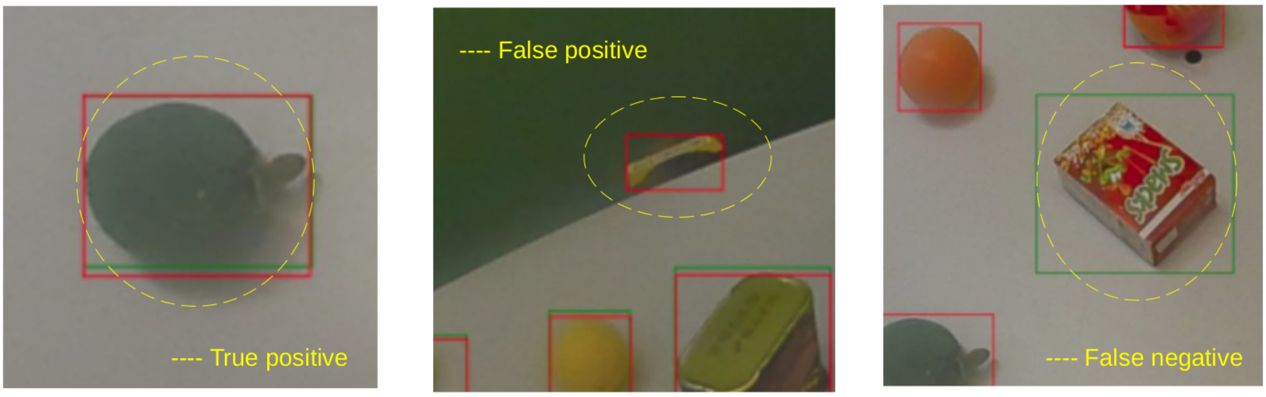


FIGURE 28 – Chosen parameters for the IoU method

The problem with these paramaters was that we had a lot of specific cases :

- Several true positives for the same object
- False positive inside an object with a high IoU score
- False negative with an IoU score different from 0

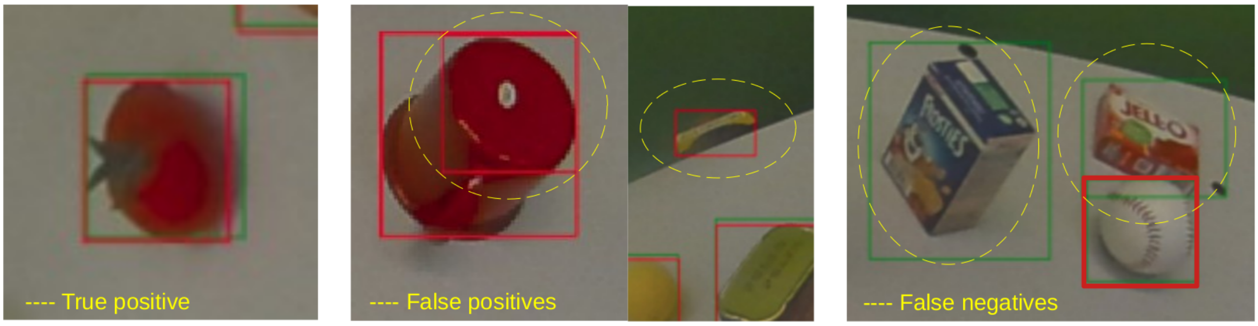


FIGURE 29 – Specific cases

Indeed, as here we evaluated the detection and not the classification, sometimes the objects detectors created two or more predicted bounding boxes for the same object. For instance, the tomato on the left image was often detected as a tomato but also as a red apple. Then, in the code we had to wrote a specific case to remove the multiple bounding boxes and to keep only one bounding box per object in order not to have wrong results.

Then, we had specific cases for the false positives. Indeed, when we have a wrong object detected like the emergency button on the picture in the middle, it is easy to say that it is a false positive because the IoU score is equal to 0. But sometimes we had false negatives inside objects so they had an high IoU score like for the cup on the middle image. In the code we had to keep only the best IoU score of each object.

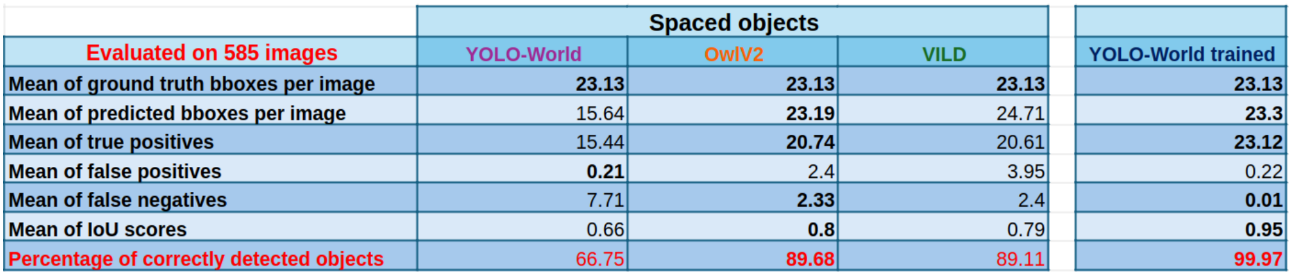
But this condition lead to another problem with another specific case : the false negatives with IoU scores different from 0. For instance, on the left image the food packaging is not detected but its IoU score is different from zero because it is close to the baseball. Then, the predicted bounding box of the food packaging touches the predicted bounding box of the baseball. Finally, in the code we also had to take this specific case into account.

6.3 The results

To compare the objects detectors, we first used the 585 images with spaced objects, then the 415 images of piled objects and finally the 1000 images with all the objects. Each time, we used YOLO-World, OwlV2 and ViLD and, as YOLO-World had a short computation time and that we had not a lot of time, we trained YOLO-World with other images of the dataset to see if the results were better or not. To compare the results, we first calculated the mean of ground-truth bounding boxes per images. To do that, we had to remove some bounding boxes because when we used YOLO-World before labeling the images, sometimes it created several bounding boxes for the same object. Then, we calculated the mean of predicted bounding boxes per image and finally we used the chosen parameters from the previous part.

6.3.1 Results on spaced objects

First, we only used the 585 images of the spaced objects and the results were surprising.



	Spaced objects			
Evaluated on 585 images	**YOLO-World**	**OwlV2**	**VILD**	**YOLO-World trained**
Mean of ground truth bboxes per image	**23.13**	**23.13**	**23.13**	**23.13**
Mean of predicted bboxes per image	15.64	**23.19**	24.71	**23.3**
Mean of true positives	15.44	**20.74**	20.61	**23.12**
Mean of false positives	**0.21**	2.4	3.95	0.22
Mean of false negatives	7.71	**2.33**	2.4	**0.01**
Mean of IoU scores	0.66	**0.8**	0.79	**0.95**
Percentage of correctly detected objects	66.75	**89.68**	89.11	**99.97**

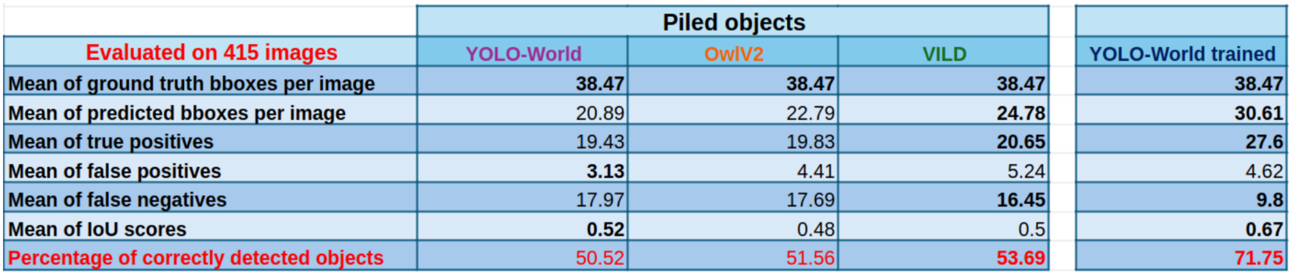
Figure 30 – Results on spaced objects

Indeed, it was with OwlV2 that we had the most correctly detected objects and the gap with the worst objects detector was of 22.93%. The worst objects detector was YOLO-World while we used it to pre-labeled the images so we might have thought that it would have been the best objects detector. Then, ViLD was pretty good while usually it is a really bad objects detector.

However this is with YOLO-World that we have fewest false positives and sometimes this is more important to have the fewest false positives than to have a high percentage of correctly detected objects. Moreover, YOLO-World was the fastest objects detector and the difference with the others is not negligible. For instance, with OwlV2, in one minute we only processed 10 images while with YOLO-World we processed 300 images. As we had not a lot of time during this four months internship, we chose to trained YOLO-World to see if we had good results. We trained it with other spaced images and we succeeded in having more true positives and the same number of false positives. The mean of IoU score was really close to 1 and we had 99.97% of correctly detected objects which is an excellent result.

6.3.2 Results on piled objects

Then, we used the 415 images of the piled objects.



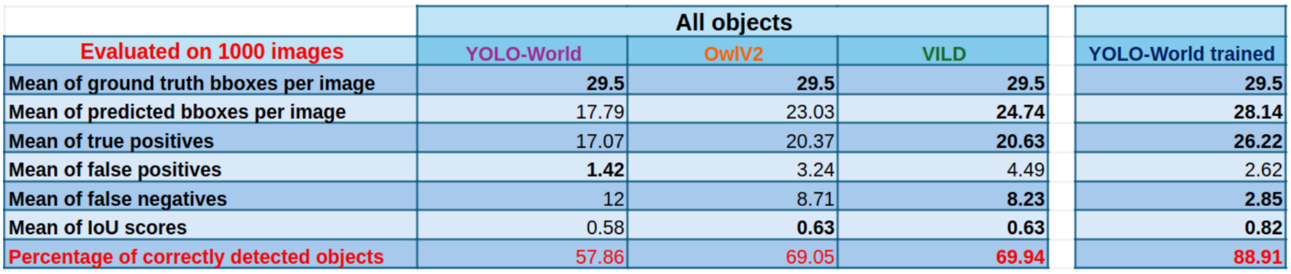
	Piled objects			YOLO-World trained
Evaluated on 415 images	YOLO-World	OwlV2	VILD	YOLO-World trained
Mean of ground truth bboxes per image	**38.47**	**38.47**	**38.47**	**38.47**
Mean of predicted bboxes per image	20.89	22.79	**24.78**	**30.61**
Mean of true positives	19.43	19.83	**20.65**	**27.6**
Mean of false positives	**3.13**	4.41	5.24	4.62
Mean of false negatives	17.97	17.69	**16.45**	**9.8**
Mean of IoU scores	**0.52**	0.48	0.5	**0.67**
Percentage of correctly detected objects	50.52	51.56	**53.69**	**71.75**

Figure 31 – Results on piled objects

Here, it was with ViLD that we had the most correctly detected objects but the gap with the worst objects detector was only of 3.17% and the percentage of correctly detected objects was much lower than with spaced objects. The worst objects detector was still YOLO-World but it remained the objects detector which whom we had fewest false positives and the fastest one. We trained again YOLO-World with other piled images and we succeeded in having more true positives. The mean of IoU score was of 0.67 which is considered as a good detection and we had 71.79% of correctly detected objects which is a good result for this kind of complicated images.

6.3.3 Results on all objects

Finally, we decided to use the 1000 images with both kind of objects.



	All objects			
Evaluated on 1000 images	YOLO-World	OwlV2	VILD	YOLO-World trained
Mean of ground truth bboxes per image	**29.5**	**29.5**	**29.5**	**29.5**
Mean of predicted bboxes per image	17.79	23.03	**24.74**	**28.14**
Mean of true positives	17.07	20.37	**20.63**	**26.22**
Mean of false positives	**1.42**	3.24	4.49	2.62
Mean of false negatives	12	8.71	**8.23**	**2.85**
Mean of IoU scores	0.58	**0.63**	**0.63**	**0.82**
Percentage of correctly detected objects	57.86	69.05	**69.94**	**88.91**

Figure 32 – Results on all objects

Here, it was still with ViLD that we had the most correctly detected objects. The gap with the worst objects detector was of 12.08% and the percentage of correctly detected objects was almost of 70%. The worst objects detector was still YOLO-World but it remained the objects detector which whom we had fewest false positives and the fastest one. We trained again YOLO-World with other images with piled and spaced objects and we succeeded in having more true positives and not a lot of false positives. The mean of IoU score was of 0.82 which is considered as a really good detection and we had almost 90% of correctly detected objects which is an excellent result for our dataset.

6.3.4 Possible explanations for the results

In terms of performance/execution time ratio, YOLO-World is the best objects detector, and even more if we do not want to have a lot of false positives in our research. Moreover, it is easy to train it to obtain really good results. But the fact that it has not the best results is surprising because we pre-labeled the images with YOLO-World so it should have had better results.

Two explanations are possible for these results. First, some objects were very close from each other like on the left image. For these two objects, the objects detector only made one predicted bounding box and then the IoU score does not mean something anymore. Then, the ground-truth bounding boxes are not perfect because we did them quickly by hand. On the right image, we can see that the IoU score of the cereal pack would be better than the one of the food can while the predicted bounding box (red) of the food can is much better. Indeed, we can see that for the food can the ground-truth bounding box (green) is really bad and that if the predicted bounding box was the yellow dotted box, its IoU score would be better than the one with the red predicted bounding box, while the red one is much better, which has no sense.

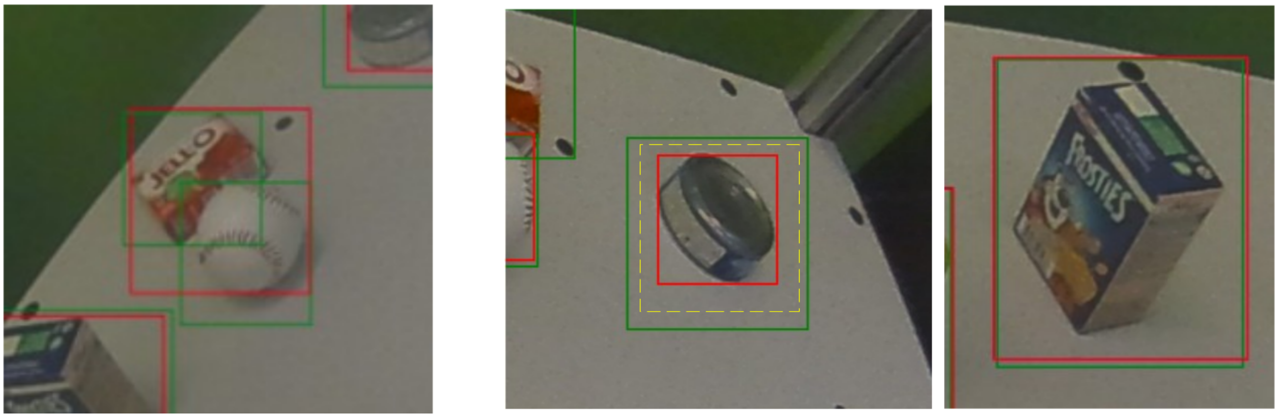


Figure 33 – Close objects and bad bounding boxes

To solve this problem, it would be useful to redraw the ground-truth bounding boxes in order to really fit the objects. To do that, we used segmentation, as we wanted to do in our initial architecture.

7 Add of SAM

As explained in the beginning of this report, we decided to use the Segment Anything Model (SAM) for the segmentation. To try to solve the two problems described above, we thought that it could be useful to use SAM before or after detection, depending of the kind of problem.

7.1 SAM after detection

First, we thought that it would be a good idea to use SAM after detection on ground-truth bounding boxes to improve the IoU results or even directly on predicted bounding boxes to improve the prediction of the detection.

For instance, on the left image, we used SAM with the green ground-truth bounding box as an input and the output is the new ground-truth bounding box in light green. We can see that the new ground-truth bounding box is better than the first one and that the IoU score will be better than before. Moreover, the fact that we used a ground-truth bounding box as an input for SAM will allow to have better results than if we used directly SAM on the whole image because here the objects are already delimited.
Then, we can also use SAM directly on predicted bounding boxes to improve the results of the objects detectors when their results are not so good like on the right image.

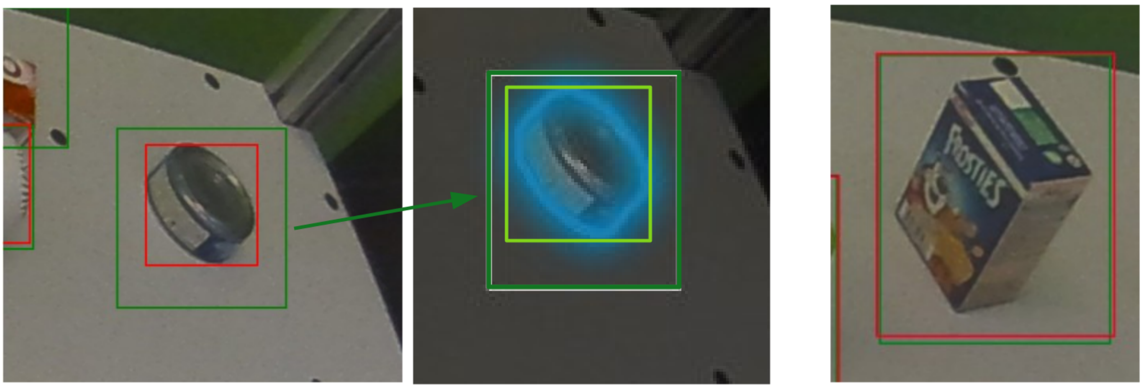


Figure 34 – Improvement of the bbox with SAM

The architecture for this use of SAM is as follows. First we use the objects detector on the image. Then, we use the predicted bounding box as an input for SAM and the segmentation allows to have the right outline of the object. Finally, the output of SAM is a more fitted bounding box.

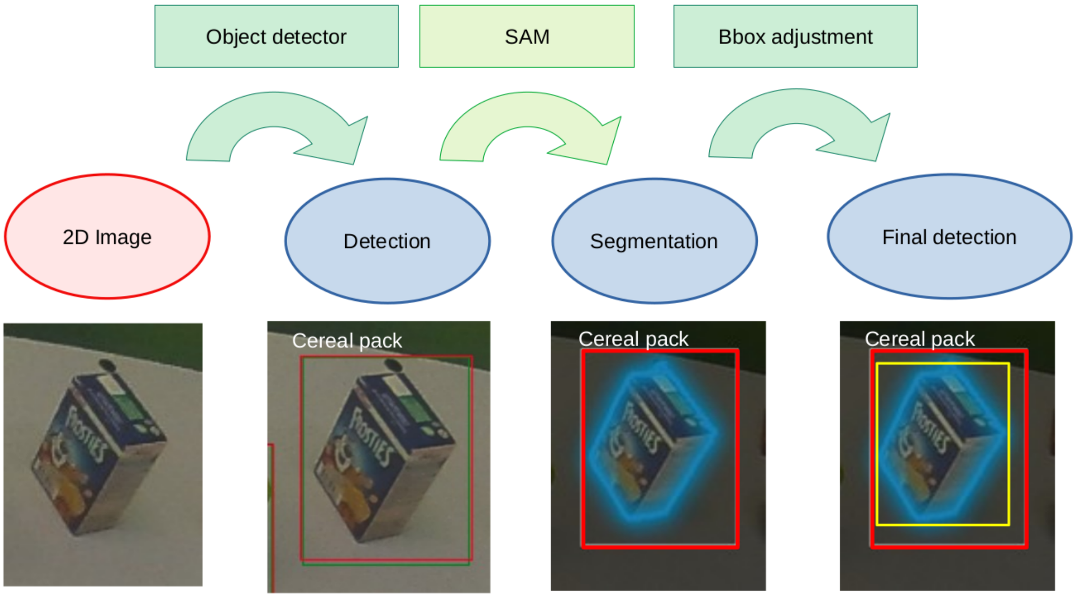


FIGURE 35 – Improved architecture

7.2 SAM before detection

Then, it could also be useful to use SAM before detection in some cases where objects are to close from each other as it was the case on the left image. On the right image, we can see that if we put directly the image into SAM, the objects are detected distinctly and that it could be easier for the objects detector to distinguish them.

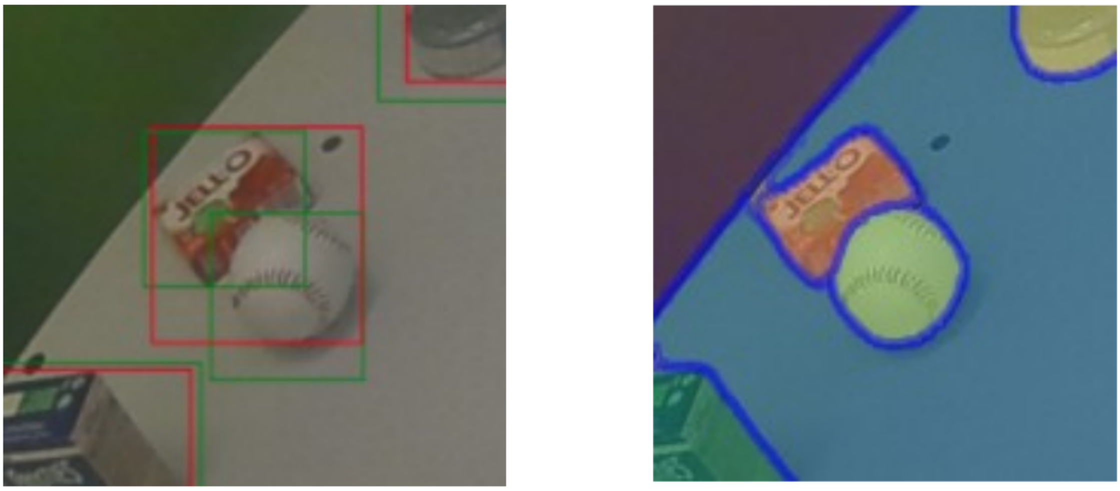


FIGURE 36 – Detection of close objects with SAM

The architecture for this use of SAM is as follows. First we use SAM to apply segmentation to the image and then, we use the objects detector to create predicted bounding boxes on the objects that were previously distinctly separated by SAM.

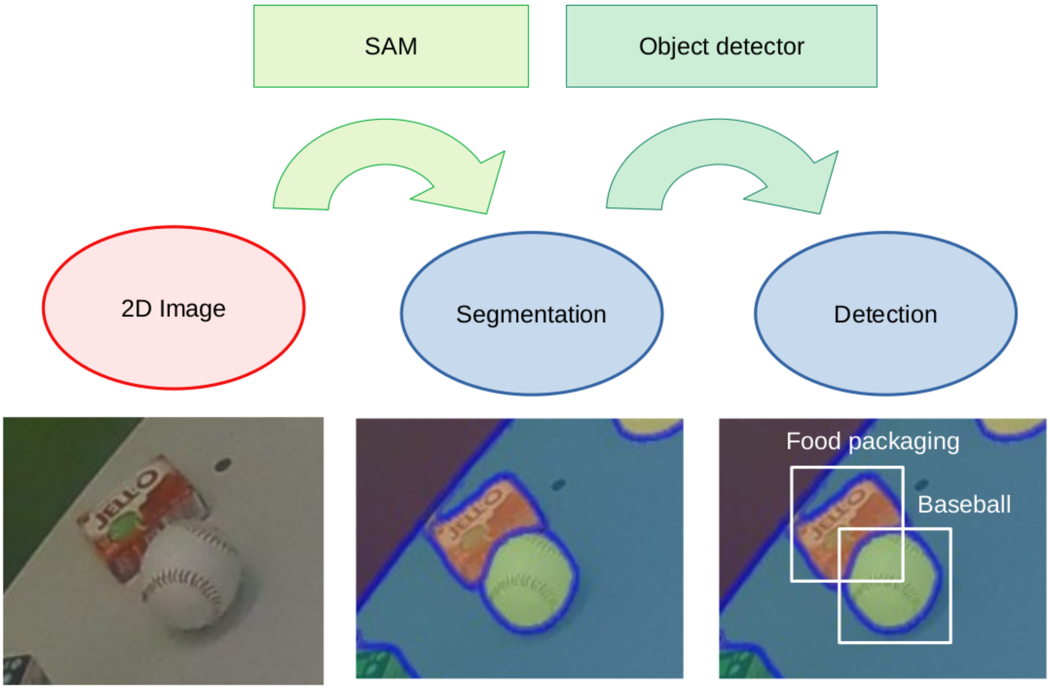


Figure 37 – Improved architecture

Unfortunately, we only had time to try these two architectures on specific images and it worked with these specific cases but we did not have time to try this on all the images to see if the results were improved or not.

8 Conclusion

8.1 Conclusion on the research topic

During my four-month internship, we succeeded in doing a lot of advances in the research topic. Indeed, even if it was impossible to cover the whole subject of our research topic, especially as the proposed architecture was innovative in the field of object detection, we worked on many parts of the subject.

First, we learnt to use the NICOL robot, both in the simulation and on the real platform. Regarding the 3D part, we realised the hardware integration of the 3D cameras and integrated them into the ROS environment. Then, we did a study about the use of several well known objects detectors on NICOL robot, which had never been done before with this robot and which will be very useful for a lot of research projects. The dataset that we collected and the labeled images will also be used by a lot of members of the WTM group. Finally, we tested the Segment Anything Model in some specific cases before and after detection.

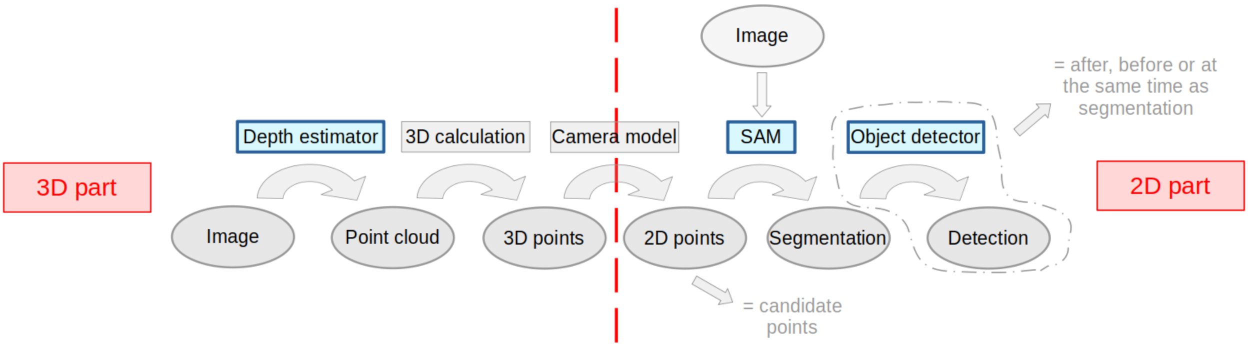


FIGURE 38 – Parts done during the internship

After I leave, it will still remain to compare the objects detectors with the use of SAM before or after the detection. Regarding the 3D part, we must add the two cameras that we removed on both sides of the table in order to have a great pointcloud and to use the code written by a member of the WTM group to obtain the 3D points and to convert them into 2D points to use the 2D part.

8.2 Conclusion on the internship

This internship was an excellent experience during which I learned many things, both professionally and personally.

It was the first time that I discovered the world of research. At first, it was a little bit complicated to find a research topic in an area I did not know but as the internship progressed, I developed more and more knowledge that allowed me to advance in the subject.

I really liked the fact that my tutor wanted me to discover each part of his work so I worked on many different things : hardware integration, algorithm development and testing, research about the state of the art and reflection on how to improve it and finally project management. Thanks to these various tasks, I could work on a lot of things, really understand how the robot works from the hardware to the software and propose an architecture for the object detection and evaluate it. Moreover, I really liked the fact that my tutor was always available while leaving me periods of autonomy because it allowed me to develop my research spirit, and try things by myself.

In addition to acquire skills in the field of object detection, I also discovered how people from different countries work together within a research group. The members of the WTM group often organised activities outside of work to gather which allows them to work better together. I found that it was very important, especially as the work of research is not always easy. Indeed, I learnt that in research there are always unexpected problems to which we must adapt quickly and which may lead to a lot of delay in work, and this is not easy mentally. This is why it was great to have such a good atmosphere in the WTM group. Finally, thanks to all the members of the group, I have always felt at ease, which allowed me to integrate well and to learn a lot during my internship, to improve my English language proficiency and presentation skills, and which gave me a great cross-cultural experience.

References

[1] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world : Real-time open-vocabulary object detection. *arXiv preprint arXiv :*, 2024.

[2] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv :2209.15639*, 2022.

[3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[4] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *arXiv preprint arXiv :2301.00764*, 2023.

[5] Meta AI Research. Segment anything. https://segment-anything.com/, 2024. Accessed : 2024-09-26.

[6] Junde Wu, Wei Ji, Yuanpei Liu, Huazhu Fu, Min Xu, Yanwu Xu, and Yueming Jin. Medical sam adapter : Adapting segment anything model for medical image segmentation, 2023.