Assignment Brief

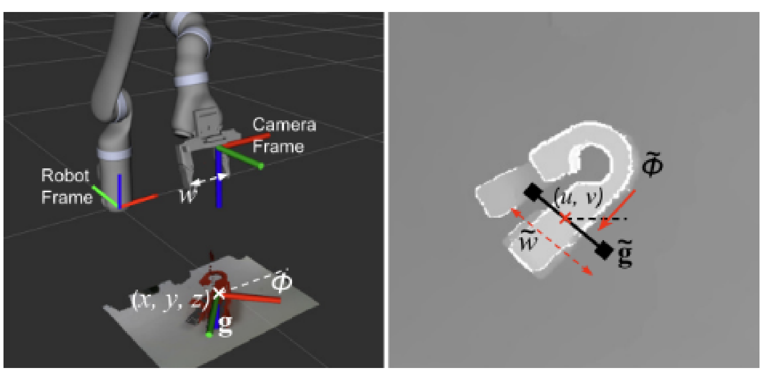
The Scenario

Imagine you are a robotic vision engineer at an innovative tech firm specializing in designing domestic service robots. These robots are the future household and office companions, assisting homeowners and office workers with chores such as cleaning, tidying, and fetching items.

For these robots to function efficiently, they need to be able to be able to perceive and understand their environment, as well as understanding how they can interact with their environment. This requires an advanced understanding of both the semantics (meaning) and geometry (structure) of their surroundings.

Your Task

Implement a robotic vision system that determines a grasp pose for objects of interest in an environment. The object of interest will be on a table top in front of the robot, but in a very cluttered scene with many other objects that should not be grasped. The grasp pose should be reported as the centroid of the gripper, width of gripper opening, and angle of gripper rotation, all relative to the image produced by the camera (see right image below).



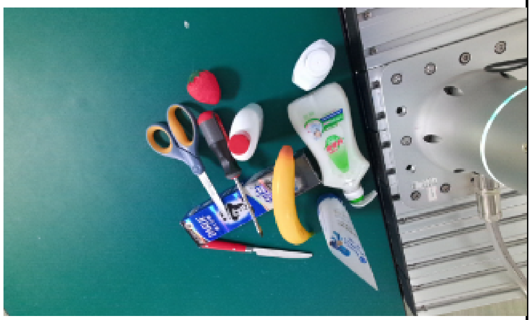
Provided Data and Code:

- **Graspnet_subset:** A folder containing all the data you will need to perform grasp prediction.
  - **validation_scenes:** A set of validation scenes that you can use to design and test the performance of your robotic vision system. You should use this data for any performance analysis you present in your oral presentation.
  - **test_scenes:** A set of test scenes that you should test your final robotic vision system on, recording and submitting these results to canvas.
- **validation_prompts.json, test_prompts.json**: The files that dictate the tests you must perform for performance evaluation and your final submission
- **validation_object_labels.json, validation_grasp_labels.json:** Files that give you ground-truth information you can use for performance evaluation
- **evaluation_support_script:** A .ipynb file that shows you how to load and visualise the data.

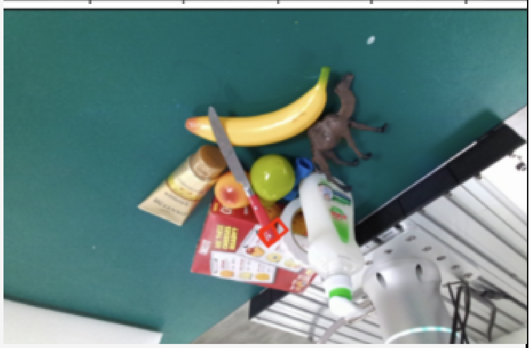
  Dataset shared with you

Data	Description

`validation_prompts.json,`
`test_prompts.json`
`{test_1: (file_name, object_name),`
`test_2: (file_name, object_name),`
`.....`
`test_n: (file_name, object_name),`
`}`
	You will be given a json file for the validation and test subset that provides a series of requested tests for you to perform. Each test contains a file name for an image in the dataset and a text prompt for the object that should be grasped. Your task will be to identify this object in the image and generate a suitable grasp pose.
	
	Objects requested will be one of the following: mug, banana, bowl, scissors, computer mouse, knife.
	
	There is the potential that a test may request an object to be grasped when that object is not actually present in the provided image.

	RGB images, corresponding to the file names in the json prompt files, that show a variety of objects in front of the robot.
	For every RGB image, there will be a corresponding depth image provided by the Kinect camera on board the robot. Depth images can be found by replacing 'rgb' with 'depth' in the file names in the prompt files. e.g. 'validation_scenes/scene_0091/kinect/**rgb**/0001.png' vs 'validation_scenes/scene_0091/kinect/**depth**/0001.png'

`validation_grasp_labels.json`
 	
`{test_1: [[[x1, y1], [x2, y2], [x3, y3], [x4, y4]],`
`...`
`],`
`test_2: [],`
`...........`
`test_n: [[[x1, y1], [x2, y2], [x3, y3], [x4, y4]]]`
`}`
	For every test in the validation_prompts.json, you will find an entry in this dictionary that lists valid grasp pose predictions. A valid grasp pose prediction is described with a rotated rectangle that represents the grasp. You can use our evaluation support script (more details below) to convert your predicted grasp into a rotated rectangle that can be compared to these ground-truth labels.

 	**Note that there are multiple valid grasps for each test (hence a list of grasps for each test is provided).** Some objects are easy to grasp and may have thousands of labelled possible grasps (e.g. top image, banana). Others are difficult to grasp, and may only have a few (e.g. bottom image, knife). An empty list will be provided when that object was not present in the image.

`validation_object_labels.json`
 	
`{test_1: (file_name, object_name, bounding box),`
`test_2: (file_name, object_name, []),`
`.........`
`test_n: (file_name, object_name, bounding box)}`
	For every test in the validation_prompts.json, you will find an entry in this dictionary that lists the bounding box for the target object. The format is as depicted, and includes the file_name, object_name (text string) and the bounding box corner coordinates as a list, e.g. [xmin, ymin, xmax, ymax]. An empty list for the bounding box will be provided when that object is not in the image.

evaluation_support_script.ipynb	This file will contain some supporting functions that will help you evaluate your model. It will show how: • validation_grasp_labels.json can be read in and grasps can be drawn on the image • validation_object_labels.json can be read in and the object bounding box can be drawn on the image

	• how to convert a grasp in (u, v, width, angle) format into a grasp rectangle prediction and draw it on an image

Expectations and Deliverables

This assessment has two deliverables: you will submit your **results** (i.e. the grasp coordinates for the objects of interest calculated by your implemented robotic vision system), and give an **oral presentation** where you will talk about your approach and analyse the performance of your system.

Quantitative Results

You have to **submit the results** of your robotic vision system as a file in JSON format, following the below structure. **You must only return the results for the test_scenes.**

```
{test_1: (u, v,  width, angle),
 test_2: 'object not present',
     .........
 test_n: (u, v, width, angle)}.
```

where:

```
u = x centroid of gripper in original image pixel coordinates
v = y centroid of gripper in original image pixel coordinates
width = width of gripper opening in original image pixel coordinates
angle = angle of gripper rotation in radians

NOTE: You must submit the string 'object not present' if you encounter a test request
for an object that is not present in an image.
```

Oral Presentation

For your presentation, you have 7 minutes to present your chosen approach and the achieved results. You should focus on:

- explaining how you designed your solution to the given problem, which algorithmic steps you chose and why
- talk about which alternatives would have been feasible, but why you decided to implement the solution you did, highlighting trade-offs or relative advantages and disadvantages of different methods
- discuss the performance of your system and any failure cases you observed

Your presentation will be followed by question time where we will ask you questions about robotic vision related to your assignment solution.

How is this Assessment Marked?

You will get marks for:

1. the performance of your submitted results
2. your oral presentation'
3. Technical Discussion shown in slide