AI Butler-Garbage Sorting

Before running the function, you need to close the App and large programs. For the closing method, refer to [4. Preparation] - [1. Manage APP control services].

1. Function Description

After the program starts, give the large model a garbage sorting instruction. After thinking, the large model will control the robotic arm to sort the garbage label code blocks on the table according to the requirements.

2. Startup

Taking the text version as an example, users with Jetson-Nano mainboard version need to enter the docker container first and then input the following command. Users with Orin mainboard can directly open the terminal and input the following command:

```
ros2 launch largemodel largemodel_control.launch.py text_chat_mode:=True
```

Then open a second terminal and input the following command:

```
ros2 run text_chat text_chat
```

Garbage sorting has the following instructions:

- Sort a single type of garbage: for example, sort toxic garbage
- Leave a certain type of garbage: for example, sort all garbage except recyclable garbage
- Sort all garbage on the desktop: for example, clean up all garbage on the desktop

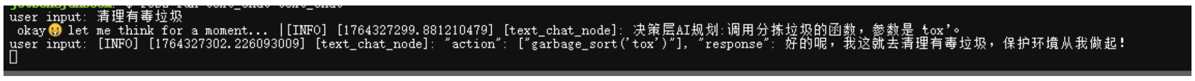
2.1. Sort a Single Type of Garbage

2.1.1. Startup

Input in the text_chat terminal:

```
Clean up toxic garbage
```

Press Enter; if it's the voice version, wake up the voice module and directly say to the voice module "Clean up toxic garbage", then wait for the large model to think and reply, as shown in the figure below:



```
user input: 清理有毒垃圾
 okay😯 let me think for a moment... |[INFO] [1764327299.881210479] [text_chat_node]: 决策层AI规划:调用分拣垃圾的函数，参数是 tox'。
user input: [INFO] [1764327302.226093009] [text_chat_node]: "action": ["garbage_sort('tox')"], "response": 好的呢，我这就去清理有毒垃圾，保护环境从我做起!
```

The robotic arm will grip the toxic garbage label code blocks on the desktop.

2.1.2. Task Planning

Call garbage_sort('tox') to sort toxic garbage.

2.1.3. Core Source Code Analysis

garbage_sort function, source code located at:
`LargeModel_ws/src/largemodel/largemodel/action_service.py`

```
def garbage_sort(self, type_):
    Arm.Arm_serial_servo_write6(90,120,10,10,90,30,1000)
    time.sleep(1.0)
    type_ = type_.strip("'\"")  # Remove single quotes and double quotes
    if type_ == "rec":
        target_type_ = float(1)
    elif type_ == "tox":
        target_type_ = float(2)
    elif type_ == "wet":
        target_type_ = float(3)
    elif type_ == "dry":
        target_type_ = float(4)
    else:
        self.get_logger().info(
            "Fatal ERROR:Incorrect type_ input,Does the AI output not meet
expectations?"
        )
        self.action_status_pub(
            "garbage_sort_failed", type_=type_
        )
        return
    # Start the robotic arm gripping program
    cmd1 = "ros2 run dofbot_pro_yolov11 yolov11_sortation"
    # Start the image conversion program
    cmd2 = "ros2 run dofbot_pro_yolov11 msgToimg"
    # Start the garbage recognition program
    cmd3 = f"ros2 run largemodel_arm yolov11_ALM --ros-args  -p target_type:=
{target_type_:.1f}"

    subprocess.Popen(
        [
            "gnome-terminal",
            "--title=grasp_desktop_garbage_sort",
            "--",
            "bash",
            "-c",
            f"{cmd1}; exec bash",
        ]
    )

    subprocess.Popen(
        [
            "gnome-terminal",
            "+--title=detect_img",
            "--",
            "bash",
            "-c",
            f"{cmd2}; exec bash",
        ]
    )

    subprocess.Popen(
        [
            "gnome-terminal",
            "--title=garbage_detect",
            "--",
            "bash",
            "-c",
            f"{cmd3}; exec bash",
        ]
    )
```

2.2. Leave a Certain Type of Garbage

2.2.1. Startup

Mainly analyze the garbage recognition program yolov11_ALM, source code path: `LargeModel_ws/src/largemodel_arm/largemodel_arm/yolov11_ALM.py`

```
# Get the garbage type to be sorted through command line arguments
self.declare_parameter('target_type', 1.0)
self.TargetType =
int(self.get_parameter('target_type').get_parameter_value().double_value)
# Modify self.waste based on the value of self.TargetType, which contains all
garbage names of this type
if self.TargetType == 1:
    self.waste = ['Newspaper','Zip_top_can','Book','Old_school_bag']
elif self.TargetType == 2:
    self.waste =
['Syringe','Expired_cosmetics','Used_batteries','Expired_tablets']
elif self.TargetType == 3:
    self.waste = ['Fish_bone','Egg_shell','Apple_core','Watermelon_rind']
elif self.TargetType == 4:
    self.waste =
['Toilet_paper','Peach_pit','Cigarette_butts','Disposable_chopsticks']

# Initialize a YOLO object detection model instance, load the pre-compiled
TensorRT engine file (best.engine), and specify the model task type as object
detection (detect)
self.yolo_model =
YOLO("/home/jetson/dofbot_pro_ws/src/dofbot_pro_yolov11/dofbot_pro_yolov11/best.
engine", task='detect')
# Create a subscriber to subscribe to the converted image data topic
self.image_sub =
self.create_subscription(ImageMsg,"/image_data",self.image_sub_callback,qos_prof
ile=1)

# In the callback function image_sub_callback
# Use YOLO11 for object detection
results = self.yolo_model(self.img, save=False, verbose=False)
boxes = results[0].boxes
# Iterate through detection results
for box in boxes:  # detections per image
    x_min, y_min, x_max, y_max = map(int, box.xyxy[0])
    class_id = int(box.cls)
    confidence = float(box.conf)
    label = f"{self.yolo_model.names[class_id]} {confidence:.2f}"
    # Calculate center position
    center_x = (x_min + x_max) // 2
    center_y = (y_min + y_max) // 2
    # Extract the information of the currently detected label code and assign it to center, including center coordinates and label code name
    center = Yolov11Detect()
    center.centerx = float(center_x)
    center.centery = float(center_y)
    center.result = str(self.yolo_model.names[class_id])
    # If the currently detected label code name is in the self.waste list, it means it is the target sorting object, publish the sorting topic flag and label code position information
    if center.result in self.waste:
        start_flag = Bool()
        start_flag.data = True
        self.pub_SortFlag.publish(start_flag)
        self.detect_flag = True
        self.cnt = self.cnt + 1
        print("Found the target.")
        self.pubDetect.publish(center)
        self.start_flag = False
```

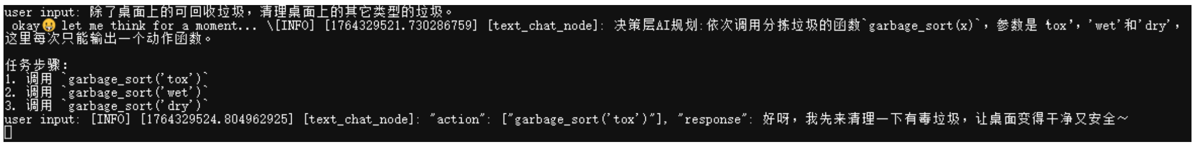
2.2. Leave a Certain Type of Garbage

2.2.1. Startup

Input in the text_chat terminal:

```
Clean up other types of garbage on the desktop except recyclable garbage.
```

Press Enter; if it's the voice version, wake up the voice module and directly say to the voice module "Clean up other types of garbage on the desktop except recyclable garbage", then wait for the large model to think and reply, as shown in the figure below:



```
user input: 除了桌面上的可回收垃圾，清理桌面上的其它类型的垃圾。
 okay😜 let me think for a moment... \[INFO] [1764329521.730286759] [text_chat_node]: 决策层AI规划:依次调用分拣垃圾的函数`garbage_sort(x)`，参数是 tox'，'wet'和'dry'，
这里每次只能输出一个动作函数。

任务步骤:
1. 调用 `garbage_sort('tox')`
2. 调用 `garbage_sort('wet')`
3. 调用 `garbage_sort('dry')`
user input: [INFO] [1764329524.804962925] [text_chat_node]: "action": ["garbage_sort('tox')"], "response": 好呀，我先来清理一下有毒垃圾，让桌面变得干净又安全~
```

Except for recyclable garbage label code blocks, the robotic arm will grip other types of garbage label code blocks on the desktop.

2.2.2. Task Planning

Call the garbage sorting function `garbage_sort(x)` in sequence, with parameters 'tox', 'wet' and 'dry'. Here only one action function can be output at a time.

Task steps:

1. Call `garbage_sort('tox')`
2. Call `garbage_sort('wet')`
3. Call `garbage_sort('dry')`

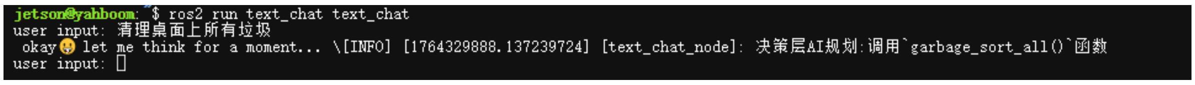
2.3. Sort All Garbage on the Desktop

2.3.1. Startup

Input in the text_chat terminal:

```
Clean up all garbage on the desktop
```

Press Enter; if it's the voice version, wake up the voice module and directly say to the voice module "Clean up all garbage on the desktop", then wait for the large model to think and reply, as shown in the figure below:



```
jetson@yahboom:~$ ros2 run text_chat text_chat
user input: 清理桌面上所有垃圾
 okay😛 let me think for a moment... \[INFO] [1764329888.137239724] [text_chat_node]: 决策层AI规划:调用`garbage_sort_all()`函数
user input: 
```

The robotic arm will sort all garbage label blocks on the desktop.

2.3.2. Task Planning

Call the garbage_sort_all() function.