HW4 Report

Team 3
游凱雯 戎宥杰 楊智翰 薛宇庭

a. Describe how you designed the program to accomplish the given task. (30%)
Ans:
The processing step of finding the principal angles and the center points:

1. Using cv2 to find the contours of an object.
2. Divide the image into connected components, then calculate the center point and principal angle for each connected component based on its momentum.

```python
def find_center(image):
    gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
    ret,thresh=cv2.threshold(gray,125,255,cv2.THRESH_BINARY)

    contours, hierarchy = cv2.findContours(thresh,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)

    block=[]
    for i,cnt in enumerate(contours):
        area = cv2.contourArea(cnt)
        # print(f'Area{i}:', area)
        if area>1000 and area<(960*1280/2):
            block.append(cnt)

    img_draw=cv2.drawContours(image,block,-1,(0,255,0),5)

    output=[]


    for i,cnt in enumerate(block):
        M = cv2.moments(cnt)
        center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))
        theta = 0.5*np.arctan2(2*M["mu11"],M["mu20"]-M["mu02"])

        output.append([i,center,theta])


    return output
```

The processing step of finding the gripper point:

1. Based on the center points, find the target points for the robotic arm gripper to move to in the image coordinates.
    a. Locate the position of the corners in the captured image to infer the center point in the real-world coordinates.

```
def ImagetoRobot(x,y):
    img_y_max=1280
    img_x_max=960
    robot_y_min=171.70
    robot_y_max=507.10
    robot_x_min=19.63
    robot_x_max=460.95

    y2=robot_y_max-x*(robot_y_max-robot_y_min)/img_x_max
    x2=y*(robot_x_max-robot_x_min)/img_y_max+robot_x_min

    return (x2,y2)
```

2. Based on the principal angle, find the orientation of the gripper. In order to make the robotic arm reach the target direction and position more quickly, adjustments will be made to the calculated rotation direction.

```
x_new,y_new = ImagetoRobot(data[1][1],data[1][0])
deg=-(data[2]*180/math.pi+90)
if deg > -90:
    deg -= 180
if deg < -270:
    deg += 180
```

3. Find the translation formula between the table and the captured picture.
4. Convert the image coordinates obtained in Step 1 into table coordinates.
5. Move the gripper to the designated pose.

a. Lower the gripper height in segments to improve stability.

```
move_to(f"{(int)(x_new)}, {(int)(y_new)}, 300, -180.00, 0.0, {deg}")
for i in range(160, 109, -10):
    move_to(f"{(int)(x_new)}, {(int)(y_new)}, {i}, -180.00, 0.0, {deg}")

set_io(1.0)
move_to(f"{(int)(x_new)}, {(int)(y_new)}, 300, -180.00, 0.0, {deg}")


move_to("571, 180, 300, -180.00, 0.0, 180.00")
move_to(f"571, 180, {height}, -180.00, 0.0, 180.00")
set_io(0.0)
move_to("571, 180, 300, -180.00, 0.0, 180.00")
height += 25
```

b. Share any challenges faced during development. (20%)
Ans:

1. It costs a lot of time to use the method from homework 3 part B to find the center and middle line.
2. When we run the program in our team's working environment, it outputs content from another team.
3. Encountering difficulties in converting image coordinates into robot coordinates.

c. Outline the solutions implemented to address these challenges. (10%)
Ans:

1. Develop another method. Using the findContours function by OpenCV to find the contour of objects.
2. The issue remained unresolved until the TA announced that there was no need to use the virtual environment anymore.
3. Try several methods. In addition, there was also the problem of inaccurate measurement. Finally, we used

the free mode of the robot arm to directly point at the four corners of the image to obtain the coordinates.

- Work distribution:
  - Report: team3
  - Experiment: team3
  - Code for detecting center points and principal angles: team3

- Demo video link :
  - https://youtu.be/GaMTFDqGVAY

- GitHub link :
  - https://github.com/Robotics-Team3-NTU2023/Assignment4