

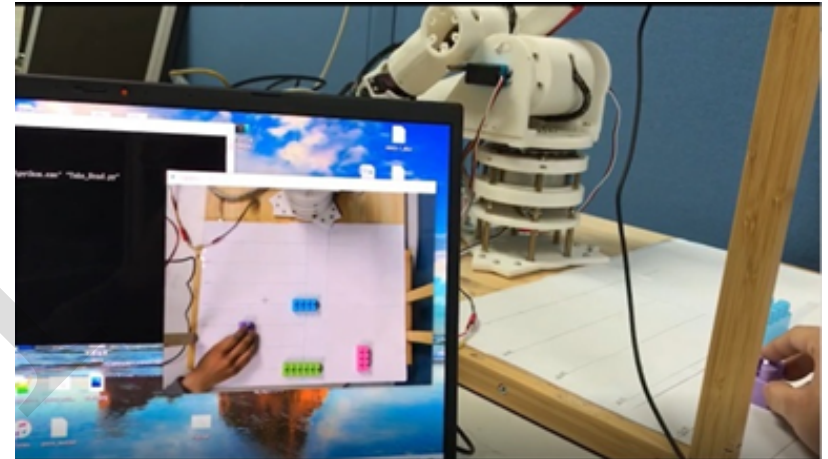
Operation Manual

Step 1:
Double-click `project.bat`(lower image) to start the YOLOv5 environment and activate the camera for real-time image capture.

```
C:\Users\al2\Desktop>echo "activate YOLOV5 Env & open camera"
"activate YOLOV5 Env & open camera"

C:\Users\al2\Desktop>cd C:\Users\al2\Desktop\Project

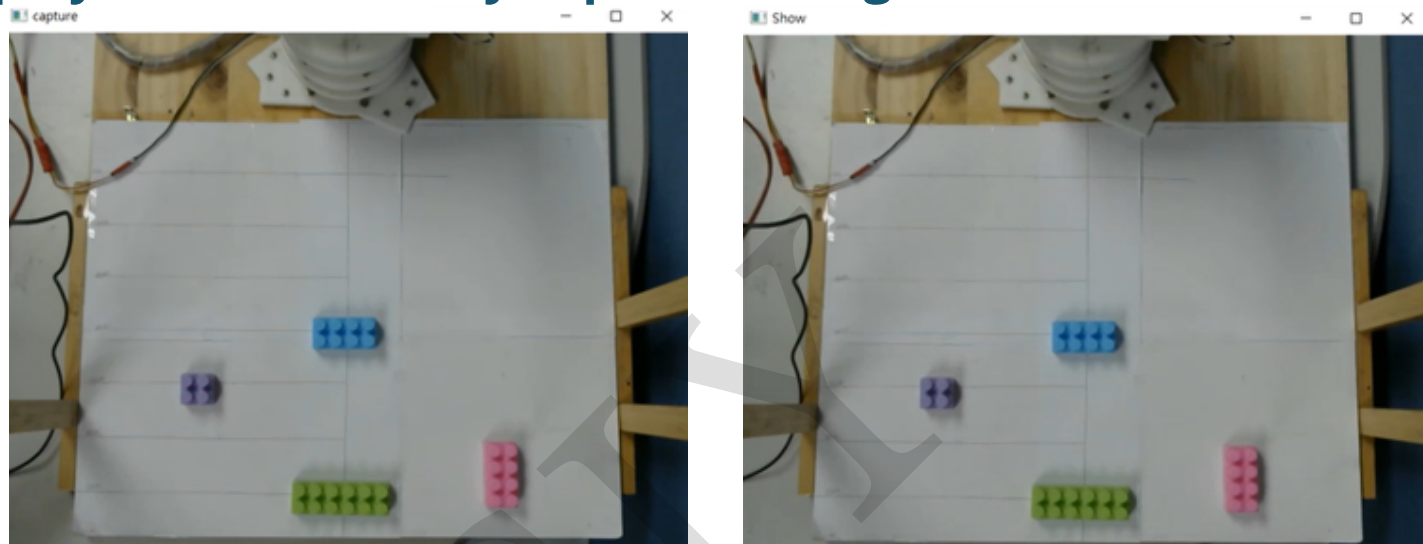
C:\Users\al2\Desktop\Project>"C:\Users\al2\anaconda3\envs\YOLOV5\python.exe" "Take_Read.py"
time1.111270 :
press s to capture image
```



```
project - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明
echo "activate YOLOV5 Env & open camera"
cd C:\Users\al2\Desktop\Project
"C:\Users\al2\anaconda3\envs\YOLOV5\python.exe" "camera_capture.py" ::2024year, Rename Take_Read.py as camera_capture.py
echo "Copy 1000.jpg"
copy C:\Users\al2\Desktop\Project\1000.jpg C:\Users\al2\Desktop\YOLO_V5\data\images
echo "run detect.py"
cd C:\Users\al2\Desktop\YOLO_V5
C:\Users\al2\anaconda3\envs\YOLOV5\python.exe detect.py --weights runs\train\exp11\weights\best.pt --img 640 --source data\images\1000.jpg
pause
cd C:\Users\al2\Desktop\YOLO_V5\runs\detect\exp4
start 1000.jpg
pause
echo "show picture"
cd C:\Users\al2\Desktop\YOLO_V5
"C:\Users\al2\anaconda3\envs\YOLOV5\python.exe" "coordinate_conversion.py"
pause
"C:\Users\al2\anaconda3\envs\YOLOV5\python.exe" "motor_movement_control.py"
echo "Copy pwm.txt"
copy C:\Users\al2\Desktop\YOLO_V5\pwm.txt C:\Users\al2\Desktop\controller\pythonEx
```

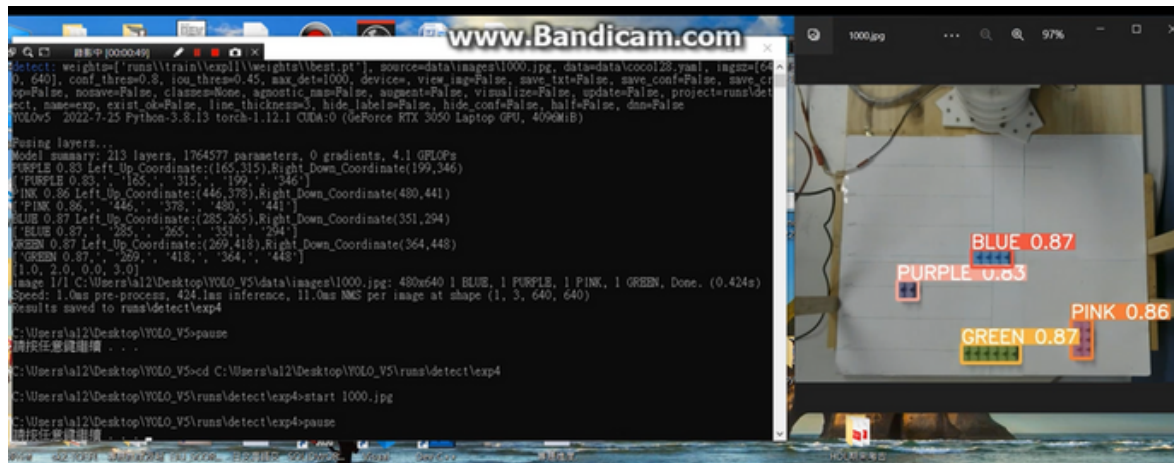
Step2:

Press 's' on the keyboard to capture an image. The "Capture" window (left) will close, and the "Show" window (right) will display the successfully captured image.



Step3:

Use the pre-trained model for image detection. In this step, the corresponding coordinates for each detected color are written to `label_coordinate.txt`. (right image)



Step4:

The user selects the target object (to be gripped). In the example, entering '3' to select GREEN.

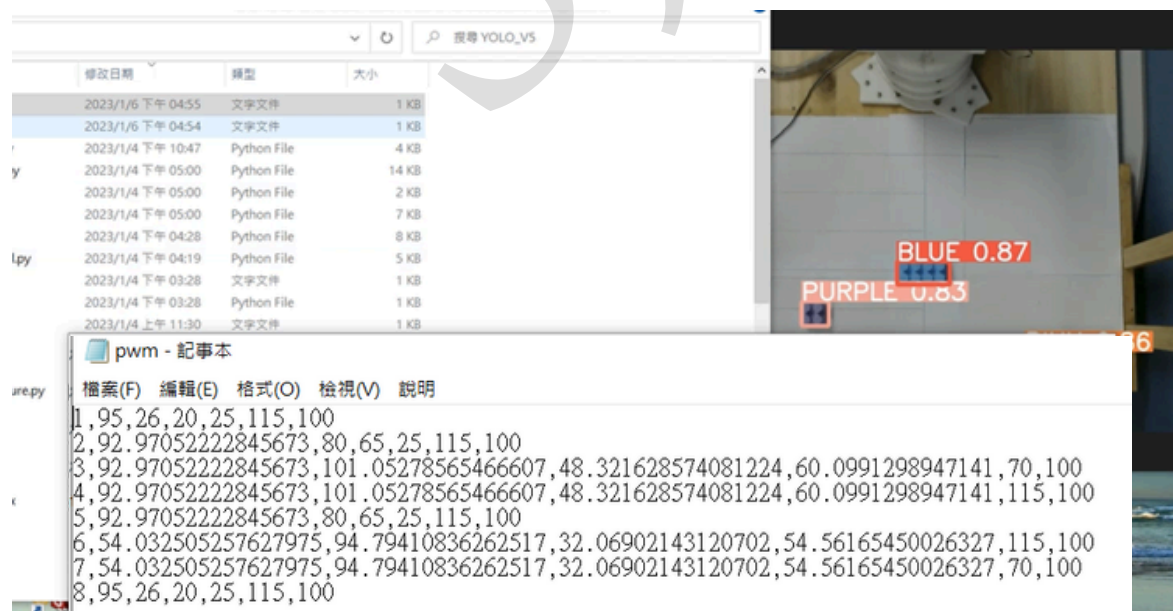
```
\\envs\\YOLOV5\\DLLs', 'C:\\Users\\ai2\\anaconda3\\envs\\YOLOV5\\lib', 'C:\\Users\\ai2\\anaconda3\\envs\\YOLOV5', 'C:\\U
ers\\ai2\\anaconda3\\envs\\YOLOV5\\lib\\site-packages', 'C:\\Users\\ai2\\anaconda3\\envs\\YOLOV5\\lib\\site-packages\\w
in32', 'C:\\Users\\ai2\\anaconda3\\envs\\YOLOV5\\lib\\site-packages\\win32\\lib', 'C:\\Users\\ai2\\anaconda3\\envs\\YOLO
V5\\lib\\site-packages\\Pythonwin']
[['PURPLE 0.83', '165', '315', '199', '346'], ['PINK 0.86', '446', '378', '480', '441'], ['BLUE 0.87', '285', '265', '35
', '294'], ['GREEN 0.87', '269', '418', '364', '448'], ['1.0', '2.0', '0.0', '3.0']]
label_num_coord : [[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]
label_num_coord : [['BLUE 0.87', '285', '265', '351', '294'], ['PURPLE 0.83', '165', '315', '199', '346'], ['PINK 0.86'
, '446', '378', '480', '441'], ['GREEN 0.87', '269', '418', '364', '448']]
After Conversion : [['BLUE 0.87', '285', '265', '351', '294'], ['PURPLE 0.83', '165', '315', '199', '346'], ['PINK 0.86', '446', '378', '480
', '441'], ['GREEN 0.87', '269', '418', '364', '448']]
slope 0.10093749999999997 intercept 0.0
slope 0.07578124999999998 intercept -2.51214793389404e-15
BLUE Center : [32.098124999999999, 21.180859374999999]
PURPLE Center : [18.370624999999993, 25.045703124999999]
PINK Center : [46.734062499999986, 31.032421874999999]
GREEN Center : [31.946718749999999, 32.813281249999999]
[[32.098124999999999, 21.180859374999999], [18.370624999999993, 25.045703124999999], [46.734062499999986, 31.032421874999999
], [31.946718749999999, 32.813281249999999]]

BLUE : 0
PURPLE : 1
PINK : 2
GREEN : 3

Input number to choose which object to be caught : 3
```

Step5:

Use inverse kinematics to calculate the coordinates of GREEN in robot coordinate system. ex: (X, Y, Z) = (32.81, -16.2, 0.35), and write the PWM values to pwm.txt.



The screenshot shows a file explorer window with a list of files. The file 'pwm - 記事本' is selected. The contents of this file are displayed in a text editor window, showing a list of coordinates and PWM values for different objects.

Object	X	Y	Z	PWM
1	95.26	20.25	115.100	
2	92.97052222845673	80.65	25.115,100	
3	92.97052222845673	101.05278565466607	48.321628574081224	60.0991298947141,70,100
4	92.97052222845673	101.05278565466607	48.321628574081224	60.0991298947141,115,100
5	92.97052222845673	80.65	25.115,100	
6	54.032505257627975	94.79410836262517	32.06902143120702	54.56165450026327,115,100
7	54.032505257627975	94.79410836262517	32.06902143120702	54.56165450026327,70,100
8	95.26	20.25	115.100	

Step6:

A window displays the robot arm's position in the coordinate system.

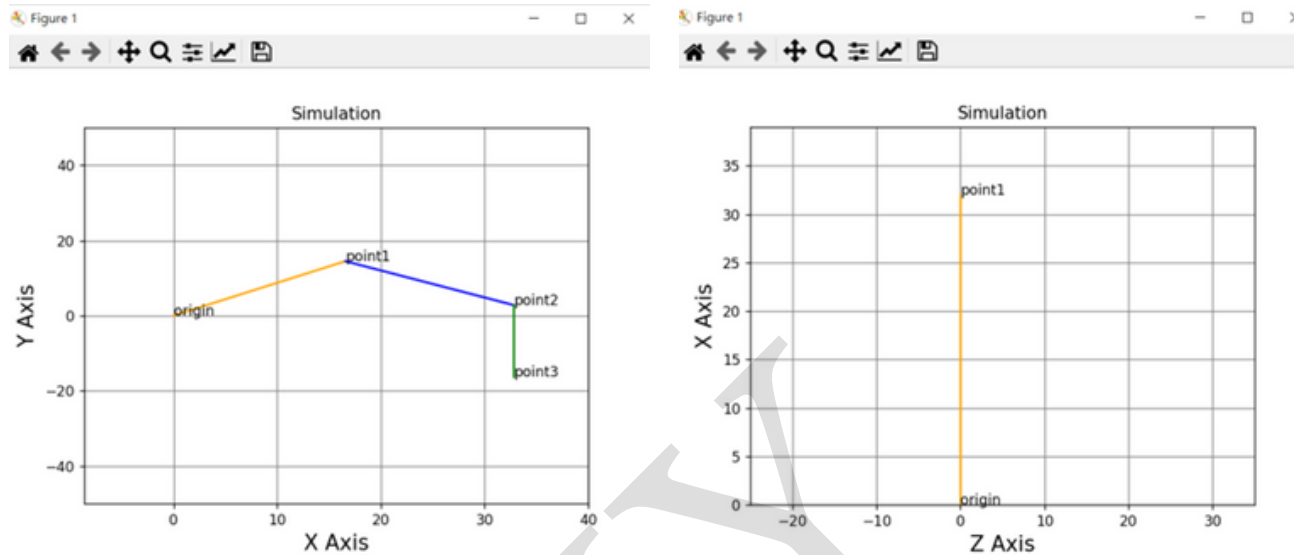


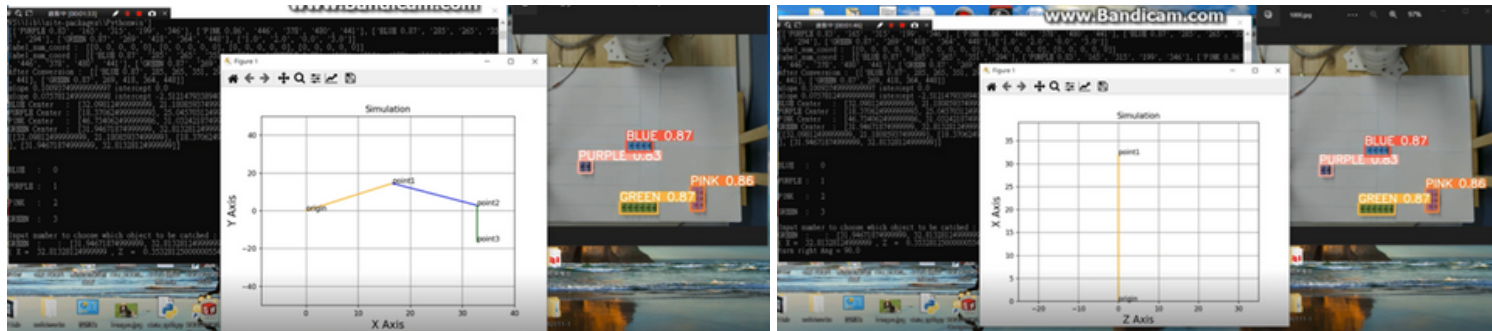
Image Description

Left: XY plane illustrating the Origin (Motor2), point1 (Motor3), point2 (Motor5), and the gripping point (Motor7).

Right: XZ plane depicting the camera's perspective of the table, highlighting the angle that Motor0 must rotate.

Upper: Enlarged view for enhanced detail and analysis.

Lower: Real-time demonstration of the robotic arm in action.



Step7: (Last Step)

PWM values are sent to the controller for motor movement.