

# 3D object recognition

## 1. Experimental purpose

Realize the car to detect cups, chairs, shoes, cameras and other objects, and draw 3D results.

## 2. Experimental path source code

Enter the car system, end the car program, enter "ip (ip is the car's ip): 8888" in the browser, enter the password "yahboom"

Then log in

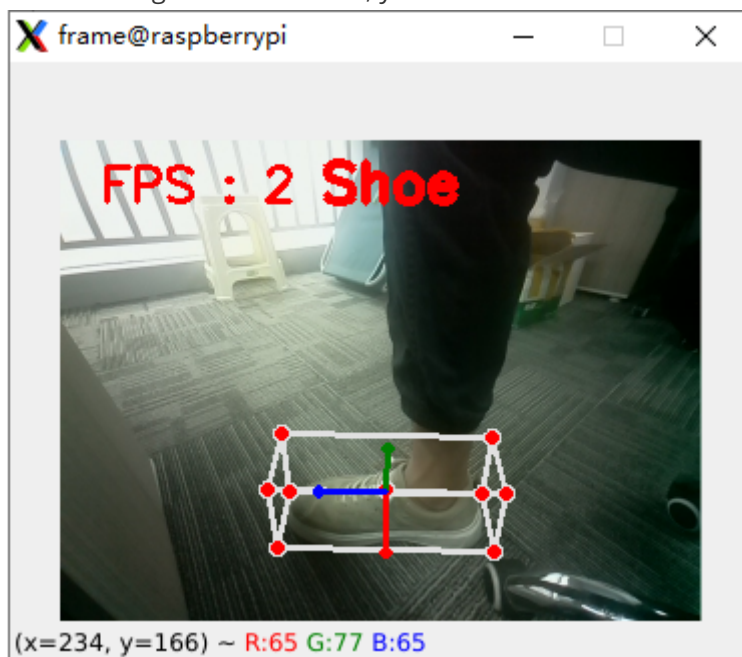
Enter the path **Rider-pi\_class/5.AI Visual Recognition Course/13. Three dimensional object recognition** and run **Objectron\_USB.ipynb**.

Or enter the command in the terminal and start the python script directly

```
cd /home/pi/Rider-pi_class/5.AI Visual Recognition Course/13. Three dimensional  
object recognition  
python3 Objectron_USB.py
```

## 3. Experimental phenomenon

After running the source code, you can see that the car detects cups, chairs, shoes, and cameras.



Press the "F" button to switch or **press the button in the upper right corner of the car screen** to switch and identify different targets. This experimental result only identifies shoes.

**Note: The F button will only have a switching effect when running this source code in the terminal. The source code running in jupyter-lab is not effective.**

## 4. Analysis of the main program source code

```
if __name__ == '__main__':
```

```

capture = cv.VideoCapture(0)
capture.set(6, cv.VideoWriter_fourcc('M', 'J', 'P', 'G'))
capture.set(cv.CAP_PROP_FRAME_WIDTH, 320)
capture.set(cv.CAP_PROP_FRAME_HEIGHT, 240)
print("capture get FPS : ", capture.get(cv.CAP_PROP_FPS))
pTime = cTime = 0
objectron = Objectron()
while capture.isOpened():
    ret, frame = capture.read()
    # frame = cv.flip(frame, 1)
    action = cv.waitKey(1) & 0xFF
    if action == ord('q'): break
    if action == ord('f') or action == ord('F') : objectron.configUP()

    # Press the button on the screen to switch
    if button.press_d():
        objectron.configUP()

    frame = objectron.findObjectron(frame)
    cTime = time.time()
    fps = 1 / (cTime - pTime)
    pTime = cTime
    text = "FPS : " + str(int(fps))
    cv.putText(frame, text, (20, 30), cv.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0,
255), 2)
    cv.imshow('frame', frame)

    #Display the image on the LCD screen
    b, g, r = cv.split(frame)
    image = cv.merge((r, g, b))
    imgok = Image.fromarray(image)
    display.ShowImage(imgok)

capture.release()
cv.destroyAllWindows()

```

From the source code analysis, it can be seen that the car will call the camera and the recognition model to determine the recognized target. If it is a target in the model, the recognition result will be displayed on the screen of the computer and the car, and the target to be recognized can be switched through the button in the upper right corner of the car.