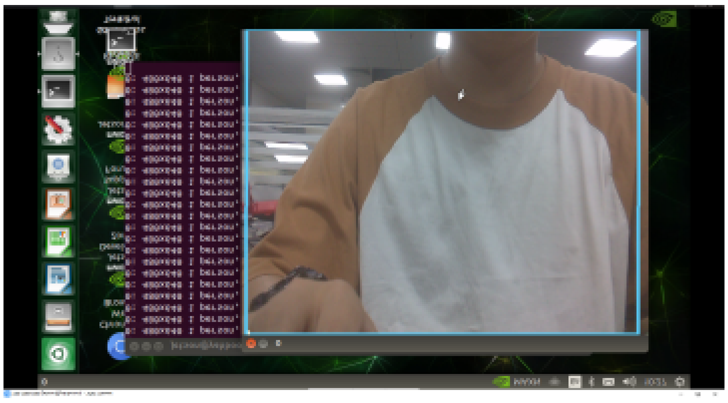
Real time detection of cameras using YOLO5

1.Usage

If you are directly using the YAHBOOM version of the mirror and using a CSI cameraRun the following command directly

```
cd ~/yolov5 && python3 detect.py --source 0
```

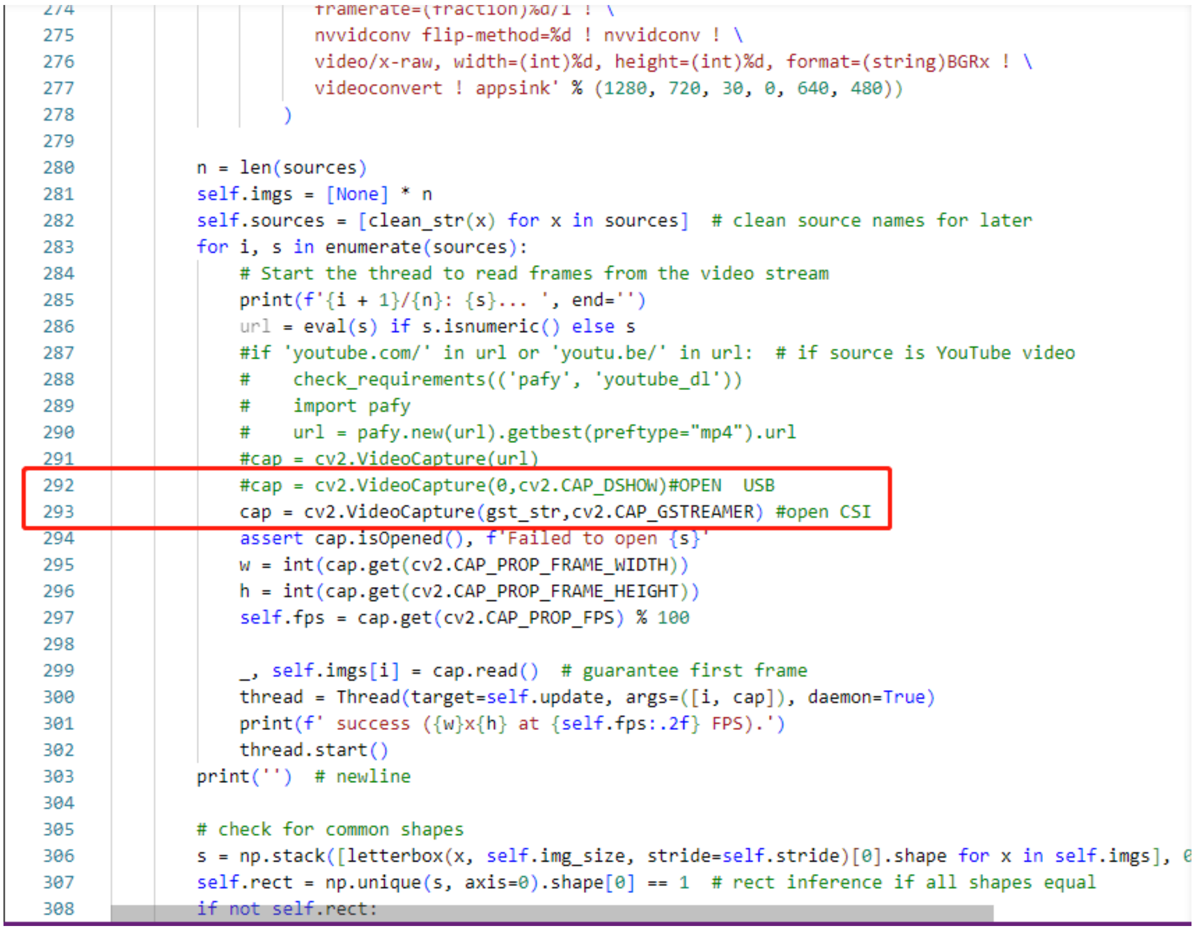
After waiting for a while, the CSI camera turned onYou can see that the screen will display the recognized object



Press Ctrl+c and turn off the camera screen to end the programAnd store the identified results in the yolov5/runs/detect/exp path (a video)2. Precautions

2.Precautions

1. If an error is reported midway due to network issues, it can be placed in the folder of yolov5 from the attachment of the environment setup, yolov5s.pt
2. If you are using a USB camera, you need to make a simple modification to the datasets. py file in~/yolov5/utils, uncomment line 292 with '#', and remove the comma and the cv2. CAP after the comma_ DSHOWS deleted. Add '#' to line 293.



```
                   framerate=(fraction)%d/1 ! \
                   nvvidconv flip-method=%d ! nvvidconv ! \
                   video/x-raw, width=(int)%d, height=(int)%d, format=(string)BGRx ! \
                   videoconvert ! appsink' % (1280, 720, 30, 0, 640, 480))
                )

        n = len(sources)
        self.imgs = [None] * n
        self.sources = [clean_str(x) for x in sources]  # clean source names for later
        for i, s in enumerate(sources):
            # Start the thread to read frames from the video stream
            print(f'{i + 1}/{n}: {s}... ', end='')
            url = eval(s) if s.isnumeric() else s
            #if 'youtube.com/' in url or 'youtu.be/' in url:  # if source is YouTube video
            #    check_requirements(('pafy', 'youtube_dl'))
            #    import pafy
            #    url = pafy.new(url).getbest(preftype="mp4").url
            #cap = cv2.VideoCapture(url)
            #cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)#OPEN  USB
            cap = cv2.VideoCapture(gst_str,cv2.CAP_GSTREAMER) #open CSI
            assert cap.isOpened(), f'Failed to open {s}'
            w = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
            h = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
            self.fps = cap.get(cv2.CAP_PROP_FPS) % 100

            _, self.imgs[i] = cap.read()  # guarantee first frame
            thread = Thread(target=self.update, args=([i, cap]), daemon=True)
            print(f' success ({w}x{h} at {self.fps:.2f} FPS).')
            thread.start()
        print('')  # newline

        # check for common shapes
        s = np.stack([letterbox(x, self.img_size, stride=self.stride)[0].shape for x in self.imgs], 0
        self.rect = np.unique(s, axis=0).shape[0] == 1  # rect inference if all shapes equal
        if not self.rect:
```