10.yolov5-lite

10.1. Introduction

YOLOv5 Lite conducts a series of ablation experiments based on YOLOv5 to make it lighter (smaller Flops, lower memory usage, fewer parameters), faster (adding shuffle channel, yolov5 head for channel pruning, and the inference speed can reach 10+FPS at least on Raspberry Pi 4B at an input_size of 320), and easier to deploy (removing the Focus layer and 4 slice operations to reduce the model quantization accuracy to an acceptable range).

Comparison of ablation experiment results:

ID	Model	Input_size	Flops	Params	Size（M)	Map@0.5	Map@.5:0.95
001	yolo-fastest	320×320	0.25G	0.35M	1.4	24.4	-
002	YOLOv5-Lite$_{e}^{ours}$	320×320	0.73G	0.78M	1.7	35.1	-
003	NanoDet-m	320×320	0.72G	0.95M	1.8	-	20.6
004	yolo-fastest-xl	320×320	0.72G	0.92M	3.5	34.3	-
005	YOLOX$_{Nano}$	416×416	1.08G	0.91M	7.3(fp32)	-	25.8
006	yolov3-tiny	416×416	6.96G	6.06M	23.0	33.1	16.6
007	yolov4-tiny	416×416	5.62G	8.86M	33.7	40.2	21.7
008	YOLOv5-Lite$_{s}^{ours}$	416×416	1.66G	1.64M	3.4	42.0	25.2
009	YOLOv5-Lite$_{c}^{ours}$	512×512	5.92G	4.57M	9.2	50.9	32.5
010	NanoDet-EfficientLite2	512×512	7.12G	4.71M	18.3	-	32.6
011	YOLOv5s(6.0)	640×640	16.5G	7.23M	14.0	56.0	37.2
012	YOLOv5-Lite$_{g}^{ours}$	640×640	15.6G	5.39M	10.9	57.6	39.1

Main features of YOLOv5-Lite:

Model compression: YOLOv5-Lite reduces the number of model parameters and computation through model pruning, quantization, knowledge distillation and other technologies, thereby reducing the demand for computing resources

Real-time performance: Due to the reduction in model size and computational complexity, YOLOv5-Lite can achieve near-real-time target detection speed on low-power devices, which is critical for real-time applications.

Adapt to edge devices: YOLOv5-Lite is designed with the limitations of edge computing in mind, allowing it to run on devices with limited computing power without sacrificing too much detection accuracy.

Flexible model size: YOLOv5-Lite provides multiple model variants, from small to large, and users can choose the appropriate model size based on the performance and power requirements of the device.

Support for multiple hardware: YOLOv5-Lite can run on a variety of hardware platforms, including but not limited to ARM architecture processors, GPUs, and dedicated AI accelerators.

Open source and community: YOLOv5-Lite inherits the open source spirit of YOLOv5, has an active community and rich resources, allowing developers to easily obtain support and customize models.

For more introduction and usage content, please refer to: https://github.com/ppogg/YOLOv5-Lite

10.2. Application of YOLOv5-Lite

Note: The following commands need to be run through vnc

Open the yolov5-lite virtual environment

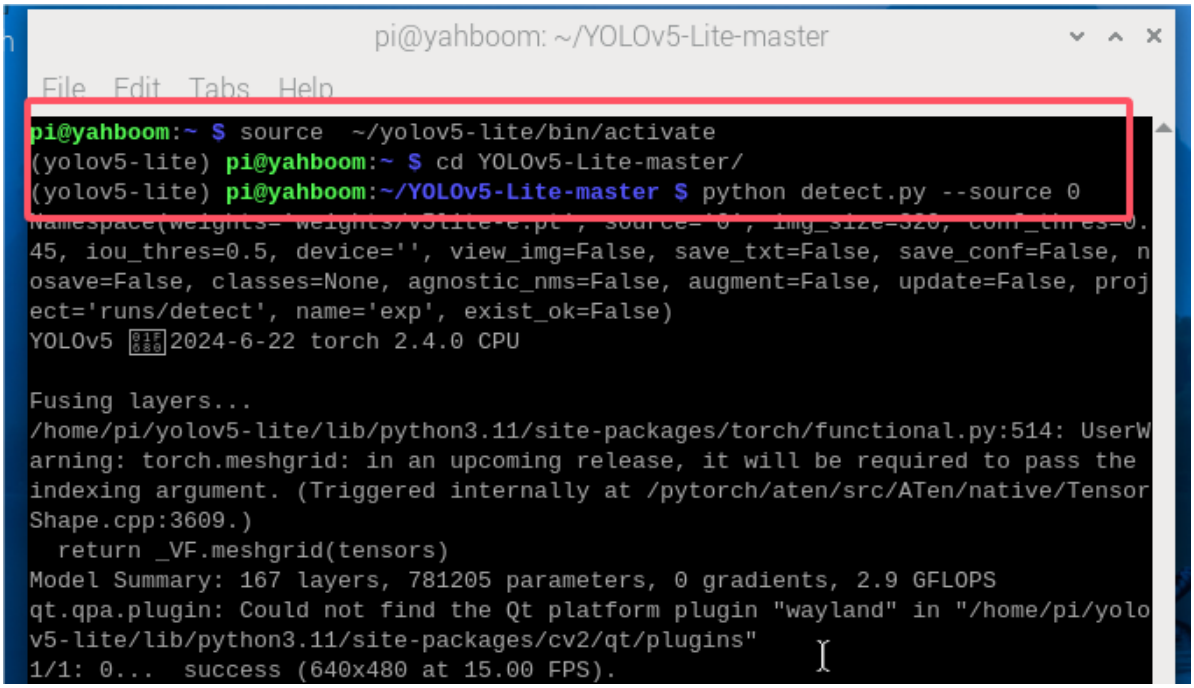
```
source ~/yolov5-lite/bin/activate
```

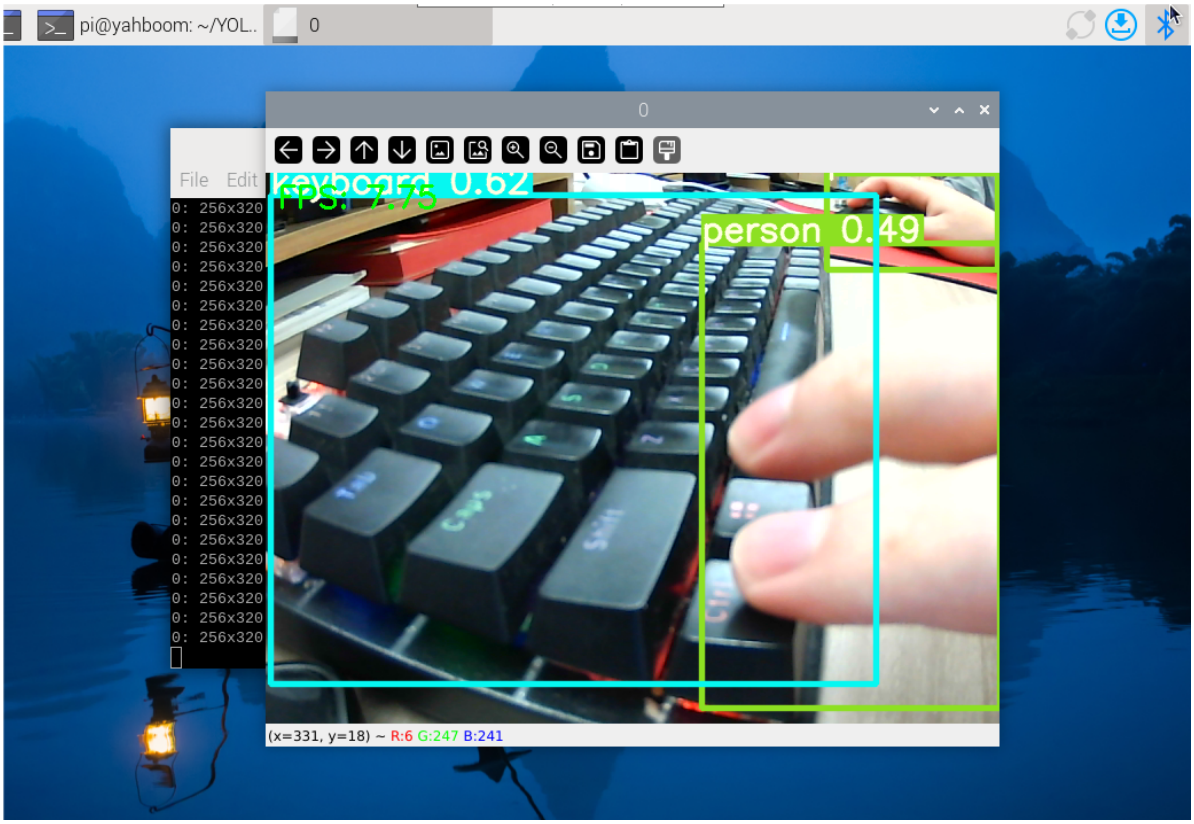
Enter the source code directory

```
cd YOLOv5-Lite-master
```

Run the sample program

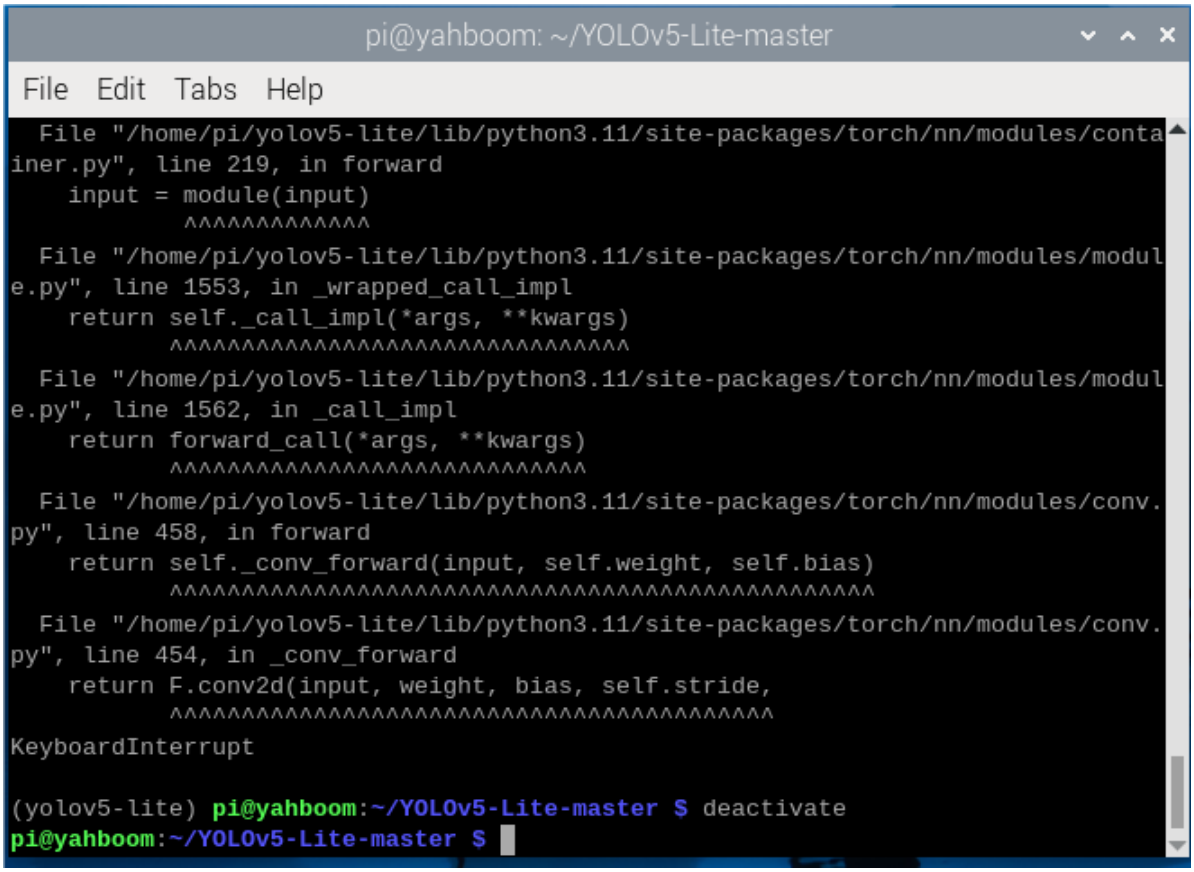
```
python detect.py --source 0
```





After the run is complete, press ctrl+c to close the program. Close the virtual environment after closing the program

```
deactivate
```


pi@yahboom: ~/YOLOv5-Lite-master
File Edit Tabs Help
File "/home/pi/yolov5-lite/lib/python3.11/site-packages/torch/nn/modules/container.py", line 219, in forward
input = module(input)
^^^^^^^^^^^^^
File "/home/pi/yolov5-lite/lib/python3.11/site-packages/torch/nn/modules/module.py", line 1553, in _wrapped_call_impl
return self._call_impl(*args, **kwargs)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/home/pi/yolov5-lite/lib/python3.11/site-packages/torch/nn/modules/module.py", line 1562, in _call_impl
return forward_call(*args, **kwargs)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/home/pi/yolov5-lite/lib/python3.11/site-packages/torch/nn/modules/conv.py", line 458, in forward
return self._conv_forward(input, self.weight, self.bias)
^^
File "/home/pi/yolov5-lite/lib/python3.11/site-packages/torch/nn/modules/conv.py", line 454, in _conv_forward
return F.conv2d(input, weight, bias, self.stride,
^^
KeyboardInterrupt
(yolov5-lite) pi@yahboom:~/YOLOv5-Lite-master $ deactivate
pi@yahboom:~/YOLOv5-Lite-master $