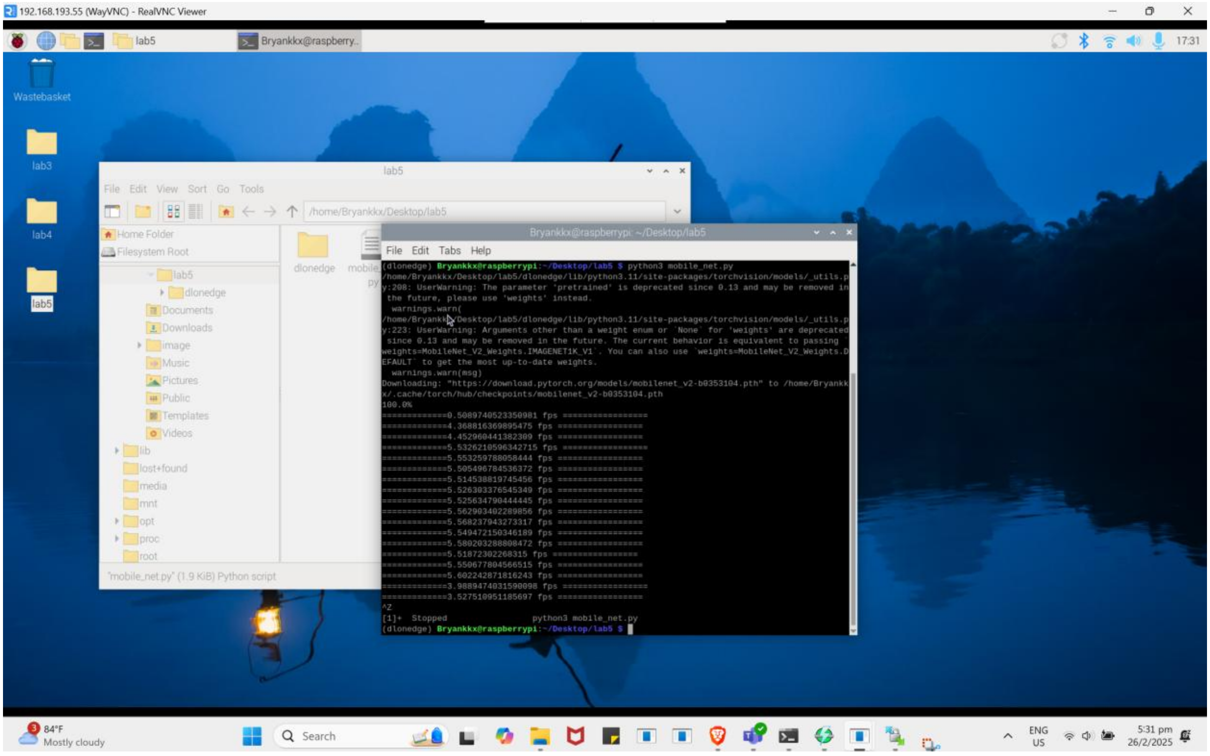
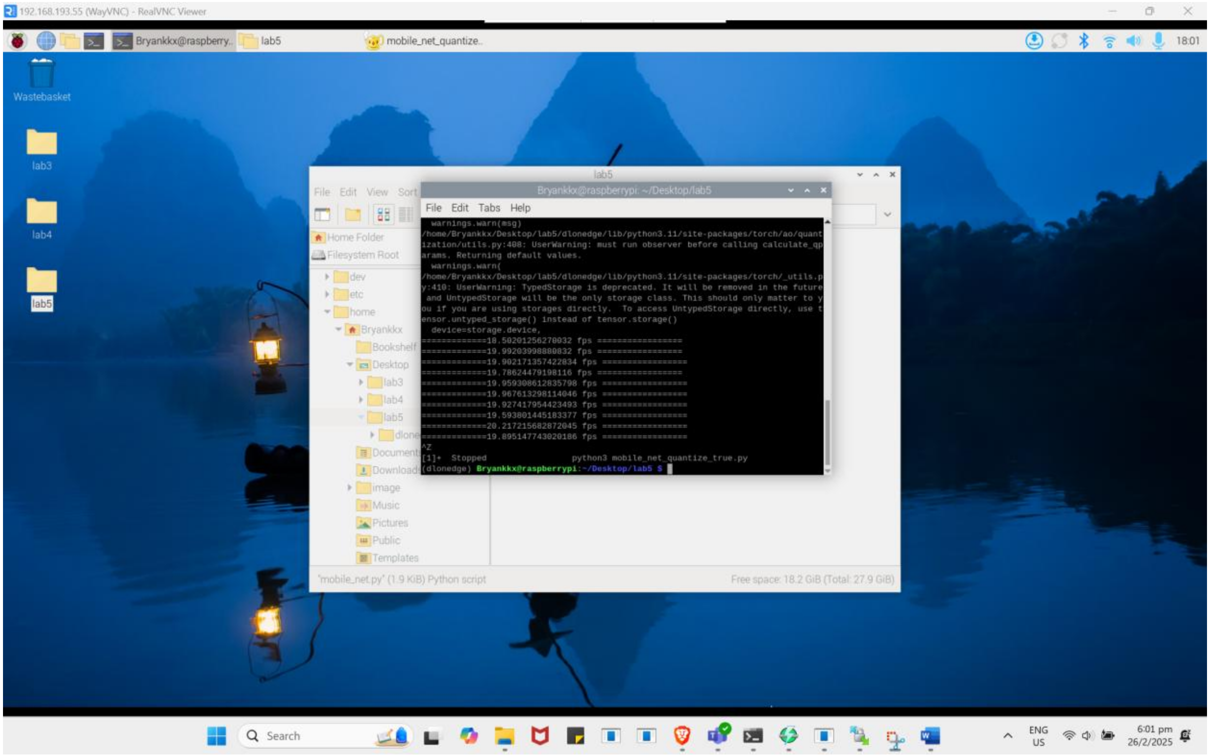
Part 1. sample code is used to directly load pre-trained MobileNetV2 model, doing model inference and finally, Observe the fps as shown in screenshot below when run on RaspberryPi 4B. As shown, with no optimization of model, we could only achieve of 5-6 fps much below our desired target.



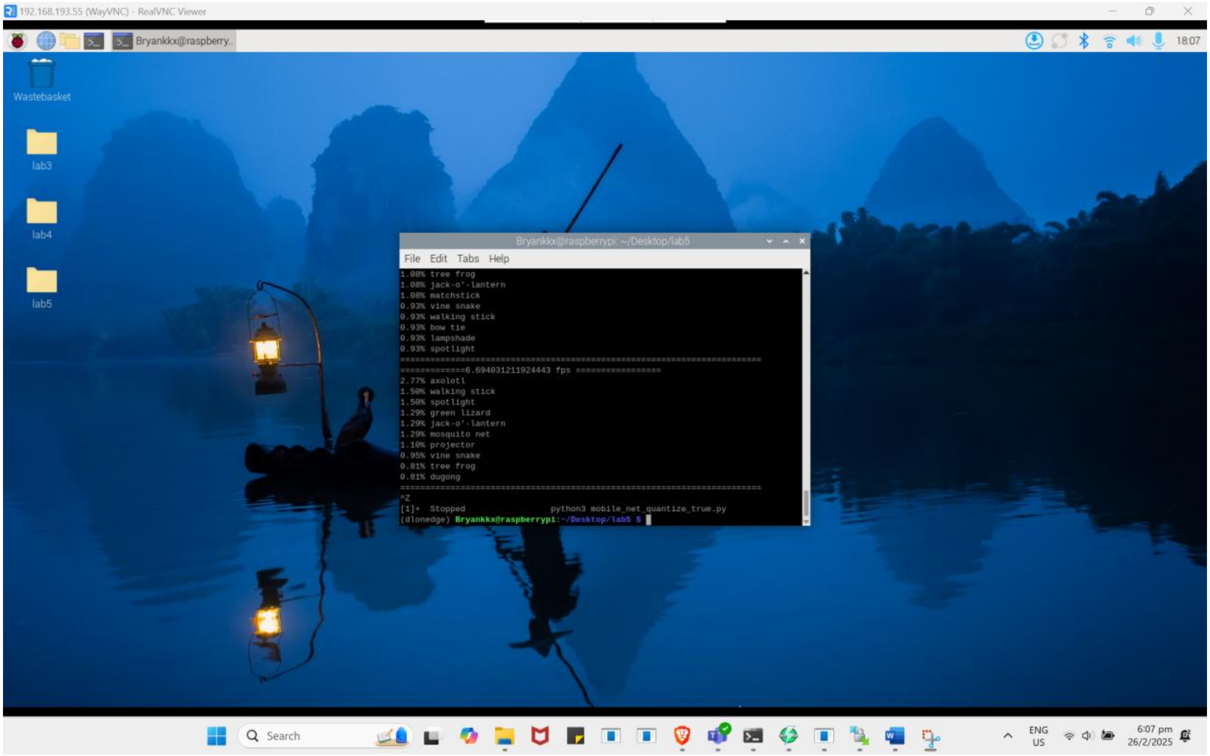
```
(dlonedge) Bryankkx@raspberrypi:~/Desktop/lab5 $ python3 mobile_net.py
/home/Bryankkx/Desktop/lab5/dlonedge/lib/python3.11/site-packages/torchvision/models/_utils.p
y:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in
 the future, please use 'weights' instead.
  warnings.warn(
/home/Bryankkx/Desktop/lab5/dlonedge/lib/python3.11/site-packages/torchvision/models/_utils.p
y:223: UserWarning: Arguments other than a weight enum or `None` for 'weights' are deprecated
 since 0.13 and may be removed in the future. The current behavior is equivalent to passing `
weights=MobileNet_V2_Weights.IMAGENET1K_V1`. You can also use `weights=MobileNet_V2_Weights.D
EFAULT` to get the most up-to-date weights.
  warnings.warn(msg)
Downloading: "https://download.pytorch.org/models/mobilenet_v2-b0353104.pth" to /home/Bryankk
x/.cache/torch/hub/checkpoints/mobilenet_v2-b0353104.pth
100.0%
=============0.5089740523350981 fps =================
=============4.368816369895475 fps =================
=============4.452960441382309 fps =================
=============5.5326210596342715 fps =================
=============5.553259788058444 fps =================
=============5.505496784536372 fps =================
=============5.514538819745456 fps =================
=============5.526303376545349 fps =================
=============5.525634790444445 fps =================
=============5.562903402289856 fps =================
=============5.568237943273317 fps =================
=============5.549472150346189 fps =================
=============5.580203288808472 fps =================
=============5.51872302268315 fps =================
=============5.550677804566515 fps =================
=============5.602242871816243 fps =================
=============3.9889474031590098 fps =================
=============3.527510951185697 fps =================
^Z
[1]+  Stopped                 python3 mobile_net.py
(dlonedge) Bryankkx@raspberrypi:~/Desktop/lab5 $ 
```

Part 2. Edit line number 11 as shown below to enable quantization in sample code to use quantized version of MobileNetV2 model.



```
  warnings.warn(msg)
/home/Bryankkx/Desktop/lab5/dlonedge/lib/python3.11/site-packages/torch/ao/quant
ization/utils.py:408: UserWarning: must run observer before calling calculate_qp
arams. Returning default values.
  warnings.warn(
/home/Bryankkx/Desktop/lab5/dlonedge/lib/python3.11/site-packages/torch/_utils.p
y:410: UserWarning: TypedStorage is deprecated. It will be removed in the future
 and UntypedStorage will be the only storage class. This should only matter to y
ou if you are using storages directly.  To access UntypedStorage directly, use t
ensor.untyped_storage() instead of tensor.storage()
  device=storage.device,
=============18.50201256278032 fps =================
=============19.99203998880832 fps =================
=============19.902171357422834 fps =================
=============19.78624479198116 fps =================
=============19.959308612835798 fps =================
=============19.967613298114046 fps =================
=============19.927417954423493 fps =================
=============19.593801445183377 fps =================
=============20.217215682872045 fps =================
=============19.895147743020186 fps =================
^Z
[1]+  Stopped                 python3 mobile_net_quantize_true.py
(dlonedge) Bryankkx@raspberrypi:~/Desktop/lab5 $ 
```

- **Part 3.** Uncomment lines 57-61 in sample code to print the top 10 predictions in real-time as shown in below video



```
1.08% tree frog
1.08% jack-o'-lantern
1.08% matchstick
0.93% vine snake
0.93% walking stick
0.93% bow tie
0.93% lampshade
0.93% spotlight
=========================================================================
=============6.694031211924443 fps =================
2.77% axolotl
1.50% walking stick
1.50% spotlight
1.29% green lizard
1.29% jack-o'-lantern
1.29% mosquito net
1.10% projector
0.95% vine snake
0.81% tree frog
0.81% dugong
=========================================================================
^Z
[1]+  Stopped                 python3 mobile_net_quantize_true.py
(dlonedge) Bryankkx@raspberrypi:~/Desktop/lab5 $ 
```