Action recognition

2.Introduction

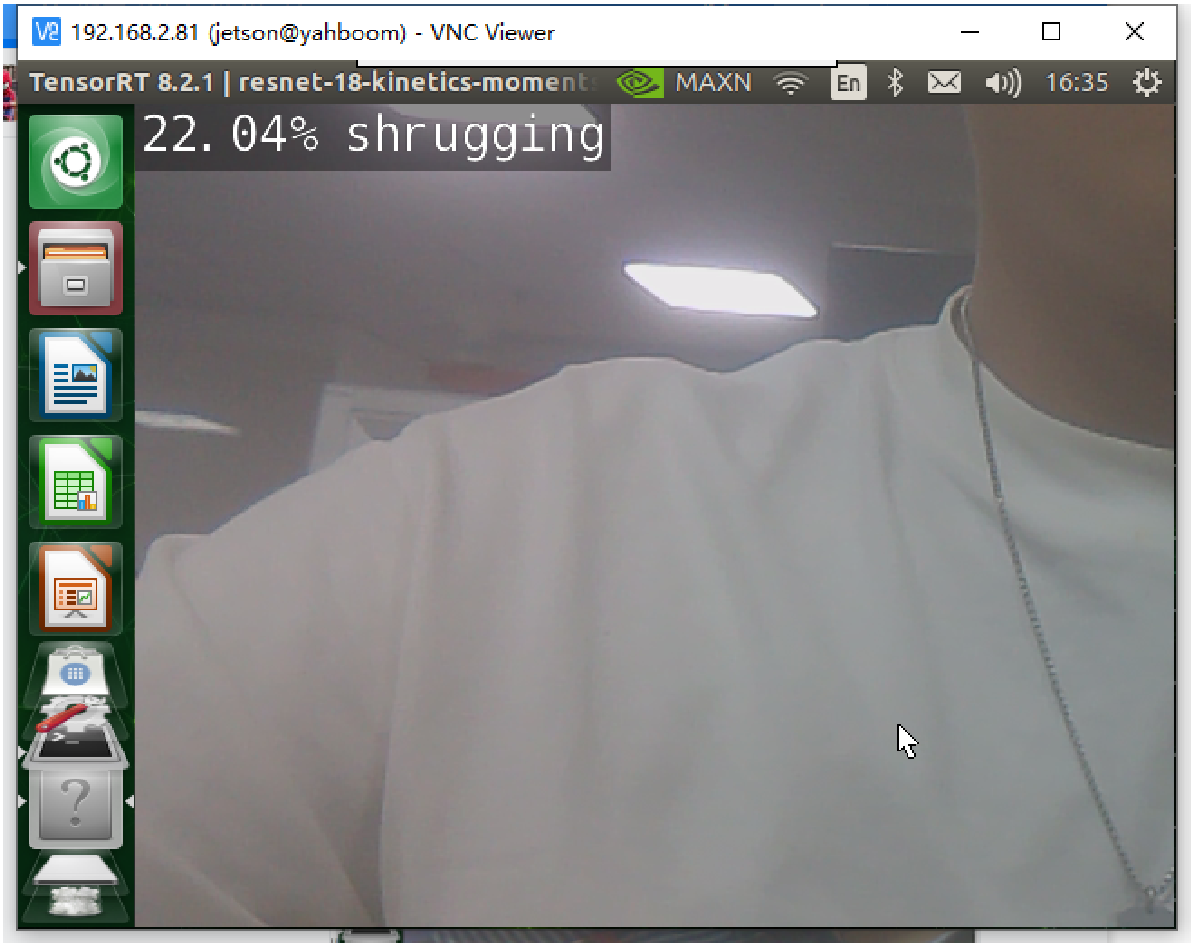
Action recognition classifies activities, behaviors, or gestures that occur in video frame sequences. DNN typically uses an image classification backbone with added time dimensions. For example, a pre trained model based on ResNet18 uses a window of 16 frames. You can also skip frames to extend the time window for the model to classify actions.The actionNet model object receives one video frame at a time, buffers them as input to the model, and outputs the class with the highest confidence. ActionNet can be used from Python and C++. As an example of using the actionNet class, the image includes sample programs in C++and Python.

2.Example of operation

To run action recognition on real-time camera streams or videos, please enter the device or file path.

```
# C++
$ ./actionnet /dev/video0           # V4L2 camera input, display output (default)
$ ./actionnet input.mp4 output.mp4  # video file input/output (mp4, mkv, avi,
flv)

# Python
$ ./actionnet.py /dev/video0           # V4L2 camera input, display output
(default)
$ ./actionnet.py input.mp4 output.mp4  # video file input/output (mp4, mkv, avi,
flv)
```



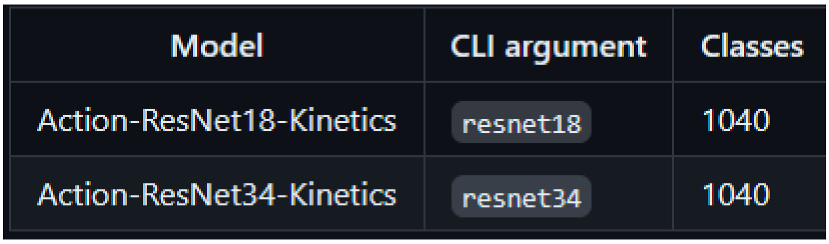
TensorRT 8.2.1 | resnet-18-kinetics-moment

22.04% shrugging

By default, the model will process every other frame to extend the time window for classifying operations.

Note: If you are building your own environment, you must download the model file to the network folder to run the above program. By using the image we provide, you can directly input the above program

The following are the available pre trained action recognition models, as well as the relevant actionnet parameters used to load them - network parameters



Model	CLI argument	Classes
Action-ResNet18-Kinetics	`resnet18`	1040
Action-ResNet34-Kinetics	`resnet34`	1040

The default value is resnet18. These models were trained on the Kinetics 700 and Moments in Time datasets (see category label list here).