Demo: [Deploy and Classify Webcam Images on NVIDIA Jetson Platform from Simulink](https://www.mathworks.com/help/coder/nvidia/ug/deploy-classify-webcam-images-jetsonTX2-from-Simulink.html)

**Prerequisites**

**Target Board Requirements**

* NVIDIA Jetson embedded platform.
* Ethernet crossover cable to connect the target board and host PC. (if you cannot connect the target board to a local network)
* USB webcam connected to the USB host port of the target.
* A monitor connected to the display port of the target.
* V4L2 and SDL (v1.2) libraries on the target.
* GStreamer libraries on the target.
* NVIDIA CUDA® toolkit and driver.
* NVIDIA cuDNN library on the target.
* Environment variables on the target for the compilers and libraries. For more information, see [Install and Setup Prerequisites for NVIDIA Boards](https://www.mathworks.com/help/coder/nvidia/ug/install-and-setup-prerequisites.html).
  + For Install and Setup, choose correct software version based on your boards. If Jetson Nano use JetPack 4.6.1- [JetPack 4.6.1 Installer](https://developer.nvidia.com/embedded/jetpack-sdk-461)
  + Choose installing JetPack using **SD Card Image Method**. Download the SD Card Image and follow the steps at [Get Started With Jetson Nano Developer Kit](https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write)

A screenshot of a computer

Description automatically generated

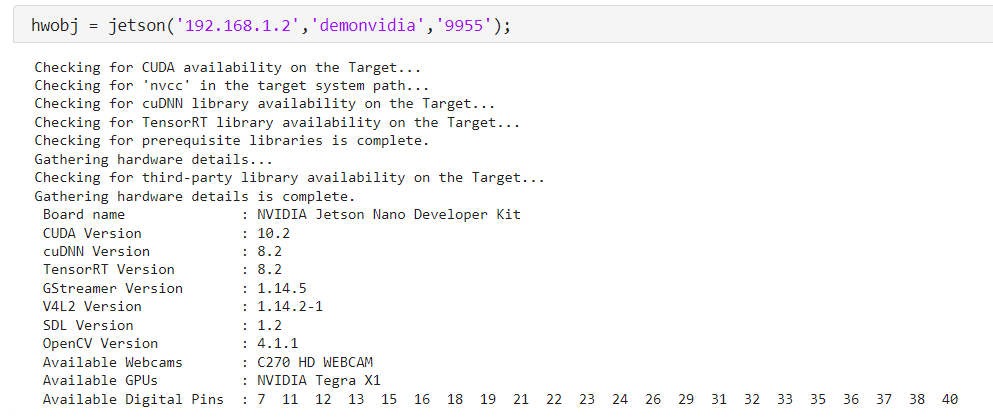
* + Next continue to finish all setup as instructed on [Install and Setup Prerequisites for NVIDIA Boards](https://www.mathworks.com/help/coder/nvidia/ug/install-and-setup-prerequisites.html)
  + Try to ping your board with your host computer.
    - Jetson address should be ###.###.###.2
    - Host address should be ###.###.###.3
    - Use direct ethernet, wifi may have firewalls
    - Disable all relevant firewalls and open port 22 for both outbound and inbound
  + If you receive error on SHH, try to consider to
    - Install openssh-client and openssh-server
      * $apt-get install openssh-client
      * $apt-get install openssh-server
      * $systemctl ssh start
      * $systemctl ssh status
    - Possible you need to regenerate ssh host keys
      * Delete ssh keys by $sudo/bin/rm-v/etc/ssh/ssh\_host\_\*
      * Regenarate by $sudo dpkg-reconfigure openssh-server
      * Then delete ssh keys in host
      * Ssh to jetson to check connection by passing >>ssh<username>@<host\_ip>

**Development Host Requirements**

* Environment variables for the compilers and libraries. For more information, see [Third-Party Hardware](https://www.mathworks.com/help/gpucoder/gs/install-prerequisites.html#mw_aa8b0a39-45ea-4295-b244-52d6e6907bff) (GPU Coder) and [Setting Up the Prerequisite Products](https://www.mathworks.com/help/gpucoder/gs/setting-up-the-toolchain.html) (GPU Coder).

**Connect to NVIDIA Jetson**

* Make sure no error/warning at this point.

****

* **Verify the GPU environment. You should not received any error**
* A screenshot of a computer

  Description automatically generated
* Identify camera connected to target board
* A screenshot of a computer

  Description automatically generated
* Set the camName and camResolution.
* Open Simulink model
* Open camera block and make sure camera name and resolution is same as in MATLAB code
* A screenshot of a computer

  Description automatically generated
* Open hardware setting. Make sure all setting is correct
* A screenshot of a computer

  Description automatically generated
* Open build options and change the build directory to /home/ubuntu. Before that make sure you have a directory in your Jetson. How to create is by
  + $cd/home
  + $mkdir ubuntu
  + $chmod ugo + rwx /home/ubuntu
  + >>exit
* Change also the display. Please check the display that available on your jetson. How to check is by
  + $echo $DISPLAY
  + If it display 1 change the display to 1, if it 0 change the display to 0
* A screenshot of a computer

  Description automatically generated
* Next step, open interface and change the deep learning target library to TensorRT instead of cuDNN. TensorRT will give a good frame per second compared to cuDNN.
* A screenshot of a computer

  Description automatically generated
* Click Apply
* Click Build, Deploy & Start
* Open home/ubuntu file in Jetson Nano, you will notice that .elf file has been created.
* A screen shot of a computer

  Description automatically generated
* Run that file on terminal
* A computer screen with text on it

  Description automatically generated