Steps for successful setup for custom model deployment on NVIDIA Jetson Nano:

1. Download SD card image for Jetson Nano Jetpack 4.6.4.
   Link: https://developer.nvidia.com/jetpack-sdk-464
2. Download and install NVIDIA Deepstream 6.0 SDK. (Download using .deb file)
   Link: https://docs.nvidia.com/metropolis/deepstream/6.0/dev-guide/text/DS_Quickstart.html
3. Clone YoloV5 - Ultralytics repo from GitHub:
   Bash Command: git clone https://github.com/ultralytics/yolov5
4. Go into yolov5 directory and then checkout to v7.0 branch for yolov5 - ultralytics repo
   Bash Command: cd yolov5
   Bash Command: git checkout v7.0
5. Update requirements.txt file in yolov5 folder:
   Bash Command: vi requirements.txt

Edit the following lines. Here you need to press i first to enter editing mode. Press ESC, then type :wq to save and quit

torch>=1.8.0

torchvision>=0.9.0

Note: torch and torchvision are excluded for now because they will be installed later.

6. Install the below dependency
   Bash Command: sudo apt install -y libfreetype6-dev
7. Install the necessary packages
   Bash Command: pip3 install -r requirements.txt
8. Install numpy 1.19.4 using following command:
   Bash Command: pip3 install --force-reinstall --no-cache-dir numpy=1.19.4
9. Install PyTorch
   First install PyTorch v1.10.0
   (Supported by JetPack 4.4 (L4T R32.4.3) / JetPack 4.4.1 (L4T R32.4.4) / JetPack 4.5 (L4T R32.5.0) / JetPack 4.5.1 (L4T R32.5.1) / JetPack 4.6 (L4T R32.6.1) with Python 3.6)
   Bash Command:
   cd ~
   sudo apt-get install -y libopenblas-base libopenmpi-dev
   wget https://nvidia.box.com/shared/static/fjtbno0vpo676a25cgvuqc1wty0fkkg6.whl -O torch-1.10.0-cp36-cp36m-linux_aarch64.whl
   pip3 install torch-1.10.0-cp36-cp36m-linux_aarch64.whl
10. Install TorchVision
    Bash Command:
    sudo apt install -y libjpeg-dev zlib1g-dev
    git clone --branch v0.11.1 https://github.com/pytorch/vision torchvision
    cd torchvision
    sudo python3 setup.py install

11. DeepStream Configuration for YOLOv5
    Bash Command:
    cd ~
    git clone https://github.com/marcoslucianops/DeepStream-Yolo
    git checkout e7c77ee9fe78d8b82caa5b01574792af429fe923
    cp DeepStream-Yolo/utils/gen_wts_yoloV5.py yolov5
    cd yolov5
    wget https://github.com/ultralytics/yolov5/releases/download/v6.1/yolov5s.pt
    python3 gen_wts_yoloV5.py -w yolov5s.pt
12. To change the inference size (default: 640) use flags as mentioned below
    -s SIZE
    --size SIZE
    -s HEIGHT WIDTH
    --size HEIGHT WIDTH
    Example for 1280:
    -s 1280
    or
    -s 1280 1280
13. Copy the generated cfg and wts files into the DeepStream-Yolo folder
    Bash Command:
    cp yolov5s.cfg ~/DeepStream-Yolo
    cp yolov5s.wts ~/DeepStream-Yolo
14. Open the DeepStream-Yolo folder and compile the library
    Bash Command:
    cd ~/DeepStream-Yolo
    CUDA_VER=11.4 make -C nvdsinfer_custom_impl_Yolo # for DeepStream 6.1
    CUDA_VER=10.2 make -C nvdsinfer_custom_impl_Yolo # for DeepStream 6.0.1 / 6.0 (Used this one in my case)
15. Edit the config_infer_primary_yoloV5.txt file according to your
    [property]
    ...
    custom-network-config=yolov5s.cfg
    model-file=yolov5s.wts
    ...
16. Edit the deepstream_app_config file

```
...
[primary-gie]
...
config-file=config_infer_primary_yoloV5.txt (CHECK THIS USUALLY THE DEFAULT VALUE IS DIFFERENT)
```

17. Change the video source in deepstream_app_config file. Here a default video file is loaded as you can see below

```
...
[source0]
...
uri=file:///opt/nvidia/deepstream/deepstream/samples/streams/sample_1080p_h264.mp4
```

18. Run the Inference
    deepstream-app -c deepstream_app_config.txt