JETSON CİHAZLAR İÇİN 6.1 JETPACK PYTORCH KURULUMU

#Guide Jetpack içeren cihazlar içindir(özellikle 6.1 için kurulum yapıcaz)

/// Aşağıdaki linkten cihazdaki mevcut JetPack sürümüne uyumlu olan torch sürümlerinden istenilen sürüm indirilir

sudo apt-get -y update;

sudo apt-get install -y python3-pip libopenblas-dev;

wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/arm64/cuda-keyring_1.1-1_all.deb

sudo dpkg -i cuda-keyring_1.1-1_all.deb

sudo apt-get update

sudo apt-get -y install libcusparselt0 libcusparselt-dev

///aşağıdaki linke giderek cihazınız için uygun jetpack versiyonuna ait pytorch linki kopyalanır ve TORCH_INSTALL e yapıştırılır

export TORCH_INSTALL=https://developer.download.nvidia.com/compute/redist/jp/v61/pytorch/torch-2.5.0a0+872d972e41.nv24.08.17622132-cp310-cp310-linux_aarch64.whl

python3 -m pip install --upgrade pip; python3 -m pip install numpy==’1.26.1’; python3 -m pip install --no-cache $TORCH_INSTALL

python3

import torch

torch.cuda.is_available()

#######Torchvision

sudo apt-get install libjpeg-dev zlib1g-dev libpython3-dev libopenblas-dev libavcodec-dev libavformat-dev libswscale-dev

///branch versiyonunu aşağıdaki linkten kontrol edebilirsiniz biz torch-2.5.0a0+872d972e41 için yapıcaz ve 2.5 için 0.20.1 gerekiyor

///https://github.com/pytorch/vision

///git clone --branch <version> https://github.com/pytorch/vision torchvision

git clone --branch v0.20.1 https://github.com/pytorch/vision torchvision

cd torchvision

export BUILD_VERSION=0.20.1

python3 setup.py install --user

cd ../

pip install 'pillow<7'

//DONE

TEST ETMEK İÇİN

python3

import torch

torch.__version__

torch.cuda.is_available()

import torchvision

torchvision.__version__

JETSON LLAMASPEAK VE JETSON CONTAINER GUIDE

sudo gedit /etc/docker/daemon.json

“default-runtime”: “nvidia”

sudo systemctl restart docker

sudo usermod -aG docker $USER
newgrp docker

wget --content-disposition https://ngc.nvidia.com/downloads/ngccli_arm64.zip && unzip ngccli_arm64.zip && chmod u+x ngc-cli/ngc
find ngc-cli/ -type f -exec md5sum {} + | LC_ALL=C sort | md5sum -c ngc-cli.md5
echo “export PATH=\”\$PATH:$(pwd)/ngc-cli\”” >> ~/.bash_profile && source ~/.bash_profile
ngc config set

ngc registry resource download-version nvidia/riva/riva_quickstart_arm64:2.17.0

cd riva_quickstart_arm64_v2.17.0

bash riva_init.sh
bash riva_start.sh

////Python Clients Testler için

pip3 install testresources

sudo apt install portaudio19-dev

pip3 install pyaudio

sudo adduser $USER audio

sudo adduser $USER pulse-access

```
newgrp pulse-access

git clone https://github.com/nvidia-riva/python-clients.git
cd python-clients
 git submodule init
 git submodule update --remote --recursive
 pip install -r requirements.txt
python3 setup.py bdist_wheel
pip install --force-reinstall dist/*.whl
pip install nvidia-riva-client
```

git clone https://github.com/dusty-nv/jetson-containers
bash jetson-containers/install.sh

If you're going to be building containers, you need to set Docker's `default-runtime` to `nvidia`, so that the NVCC compiler and GPU are available during `docker build` operations. Add `"default-runtime": "nvidia"` to your `/etc/docker/daemon.json` configuration file before attempting to build the containers:

```
{
   "runtimes": {
      "nvidia": {
         "path": "nvidia-container-runtime",
         "runtimeArgs": []
      }
   },

   "default-runtime": "nvidia"
}
```

```
jetson-containers run --env HUGGINGFACE_TOKEN=hf_xyz123abc456 \ $(autotag nano_llm) \ python3 -m nano_llm.agents.web_chat --api=mlc \ --model meta-llama/Meta-Llama-3-8B-Instruct \ --asr=riva --tts=piper
```