

AIPro iNet Solution Guide for Ubuntu Nvidia GPU v1.1 (English)

- **Document version:** v1.1
- **Date:** 2024-11-19
- **Author:** [HaHV](#)

General information

Source code and documents

- Link: [AIProCo/iNet-API-Demo](#)
- This guild file could found in iNet source code at [docs](#)

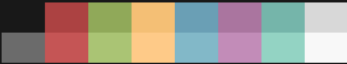
Dev environments (envs)

- Ubuntu 22.04 LTS
 - You can install Ubuntu using this [guide](#)
 - You can install Nvidia driver using this [guide](#)
- Cuda 12.1.0
- cuDNN 8.9.6
- OpenCV-4.9.0 (included)
- TensorRT-8.6.1.6 (included)
- OpenVINO 2023.2 (included)
- **GCC 13** (C++ complier) **required** due to iNet used `std::format` in C++ 20 standards
- C++ build tool **CMake** with minimum VERSION 3.10 (tested with version 3.22.1 in this guide)

The following image is the test environment for this guide:

```
> neofetch
      .-/+00SSSS00+/- .
      `:+SSSSSSSSSSSSSSSS+:`
      -+SSSSSSSSSSSSSSSSyySSSS+-
      .OSSSSSSSSSSSSSSSSdMMMMNySSSSO.
      /SSSSSSSSSSShdmmNNmmyNMMMMhSSSSSS/
      +SSSSSSSSShmydMMMMMMMMNddddySSSSSSSS+
      /SSSSSSSSShNMMMyhhyyyhmNMMMNhSSSSSSSS/
      .SSSSSSSSdMMMNhSSSSSSSSShNMMMdSSSSSSSS.
      +SSSSShhhyNMMNySSSSSSSSSSSyNMMMySSSSSSSS+
      OSSyNMMMNyMMhSSSSSSSSSSShmmhSSSSSSSSO
      OSSyNMMMNyMMhSSSSSSSSSSShmmhSSSSSSSSO
      +SSSSShhhyNMMNySSSSSSSSSSSyNMMMySSSSSSSS+
      .SSSSSSSSdMMMNhSSSSSSSSShNMMMdSSSSSSSS.
      /SSSSSSSSShNMMMyhhyyyhdNMMMNhSSSSSSSS/
      +SSSSSSSSdmydMMMMMMMMNddddySSSSSSSS+
      /SSSSSSSSShdmmNNmmyNMMMMhSSSSSS/
      .OSSSSSSSSSSSSSSSSdMMMMNySSSSO.
      -+SSSSSSSSSSSSSSSSyySSSS+-
      `:+SSSSSSSSSSSSSSSS+:`
      .-/+00SSSS00+/- .

ha@ha-Sword-17-A11UE
-----
OS: Ubuntu 22.04.5 LTS x86_64
Host: Sword 17 A11UE REV:1.0
Kernel: 6.8.0-48-generic
Uptime: 6 mins
Packages: 2011 (dpkg), 10 (snap)
Shell: zsh 5.8.1
Resolution: 1920x1080, 1080x1920
DE: GNOME 42.9
WM: Mutter
WM Theme: Adwaita
Theme: Yaru-dark [GTK2/3]
Icons: Yaru [GTK2/3]
Terminal: alacritty
CPU: 11th Gen Intel i7-11800H (16) @ 4.600GHz
GPU: NVIDIA GeForce RTX 3060 Mobile / Max-Q
GPU: Intel TigerLake-H GT1 [UHD Graphics]
Memory: 5930MiB / 15692MiB
```



Ubuntu Software Other Software Updates Authentication Additional Drivers Developer Options Ubuntu Pro

NVIDIA Corporation: GA106M [GeForce RTX 3060 Mobile / Max-Q]

This device is using the recommended driver.

Using NVIDIA driver metapackage from nvidia-driver-550 (proprietary, tested)

Using NVIDIA driver (open kernel) metapackage from nvidia-driver-545-open (proprietary)

Using NVIDIA driver metapackage from nvidia-driver-470 (proprietary)

Using NVIDIA Server Driver metapackage from nvidia-driver-535-server (proprietary)

Using NVIDIA driver (open kernel) metapackage from nvidia-driver-535-open (proprietary)

Using NVIDIA driver (open kernel) metapackage from nvidia-driver-535-server-open (proprietary)

Using NVIDIA driver (open kernel) metapackage from nvidia-driver-550-open (proprietary)

Using NVIDIA driver metapackage from nvidia-driver-545 (proprietary)

Using NVIDIA Server Driver metapackage from nvidia-driver-470-server (proprietary)

Using NVIDIA driver metapackage from nvidia-driver-535 (proprietary)

Using X.Org X server – Nouveau display driver from xserver-xorg-video-nouveau (open source)

Technical Guides

Notice:

- All paths in this guide is relative path to **iNet-API-Demo** folder.
- BEFORE running a script file (.sh), make sure it allowed to run by execute this command: `sudo chmod +x script.sh`
- when run CMake, make sure you specify gcc-13: `cmake -S . -B build -DCMAKE_C_COMPILER=/usr/bin/gcc-13 -DCMAKE_CXX_COMPILER=/usr/bin/g++-13 ...<other_args>...`

Step 1 : Install C++ compiler GCC 13 and dependencies

- [original guide](#)
- run these cmd:

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
sudo apt-get update
sudo apt-get install gcc-13 g++-13
# test
gcc-13 --version
```

- Install dependent libs: note that tbb (Intel oneAPI Threading Building Blocks) may not be installed by default, so you need to install it manually.

```
sudo apt install cmake libtbb2 g++ wget unzip ffmpeg libgtk2.0-dev
libavformat-dev libavcodec-dev libavutil-dev libswscale-dev libtbb-dev
libjpeg-dev libpng-dev libtiff-dev
```

Step 2: Install Cuda 12.1 for Ubuntu 22.04

- Install Cuda by following this official video [CUDA Tutorials I Installing CUDA Toolkit on Windows and WSL - YouTube](#) with the `.run` file downloaded from [CUDA Toolkit Archive](#).
 - Note that if we already install Nvidia driver (using [Additional Drivers](#) in Ubuntu), we should **NOT** install Nvidia driver again when installing Cuda Toolkit.
- Add Cuda to system `PATH` and `LD_LIBRARY_PATH`

```
export PATH=/usr/local/cuda-12.1/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-12.1/lib64:$LD_LIBRARY_PATH
```

Step 3: Install cuDNN 8.9.6 using Tar file

- Install `cuDNN` using `.tar` file with this [guide](#) where `.tar` file can be downloaded at [cuDNN Archive](#)

Choose [Download cuDNN v8.9.6 \(November 1st, 2023\)](#), for `CUDA 12.x` and download `cuDNN Library for Linux x86_64 (Tar)`.

```
tar -xvf cudnn-linux-x86_64-8.9.6.50_cuda12-archive.tar.xz
sudo cp cudnn-*-archive/include/cudnn*.h /usr/local/cuda-12.1/include
sudo cp -P cudnn-*-archive/lib/libcudnn* /usr/local/cuda-12.1/lib64
sudo chmod a+r /usr/local/cuda/include/cudnn*.h /usr/local/cuda-12.1/lib64/libcudnn*
```

Step 4: Download and extract necessary files

- Download and unzip one of the followings zip files. Each zip files has 3 directories after unzip: `bin`, `inputs`, and `videos`. Then, copy and paste `bin` directory to the solution directory (the directory including the `.sln` file). Move `inputs` and `videos` to the `bin` directory .
 - Cuda compute capability of your GPU should be 8.6 (RTX-30xx) or 8.9 (RTX-40xx):
 - RTX-30xx
 - RTX-40xx

Step 8: Build and run iNet framework on WSL

NOTE: If you want to build for CPU, the flag in `CMakeLists.txt` must be turn ON set (`BUILD_FOR_CPU ON`)

- Check and modify the contents of the `CMakeLists.txt` files if needed. Make sure configs correct, especially for *header file directories* and *lib file directories*.
- Run `build.sh [-force] [-debug]` to build/rebuild from scratch or enable debug mode.
 - Note that the executable file (`client`) are automatically copied to `bin` folder after build process.
- Make sure `inputs` folder (with `config.json` and `.net` mode files) and `videos` folder are moved/copied into `bin` folder.
 - Run `client`:
 - `cd bin`
 - `./client`