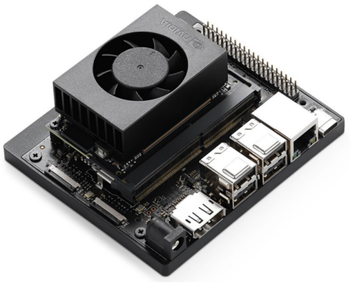
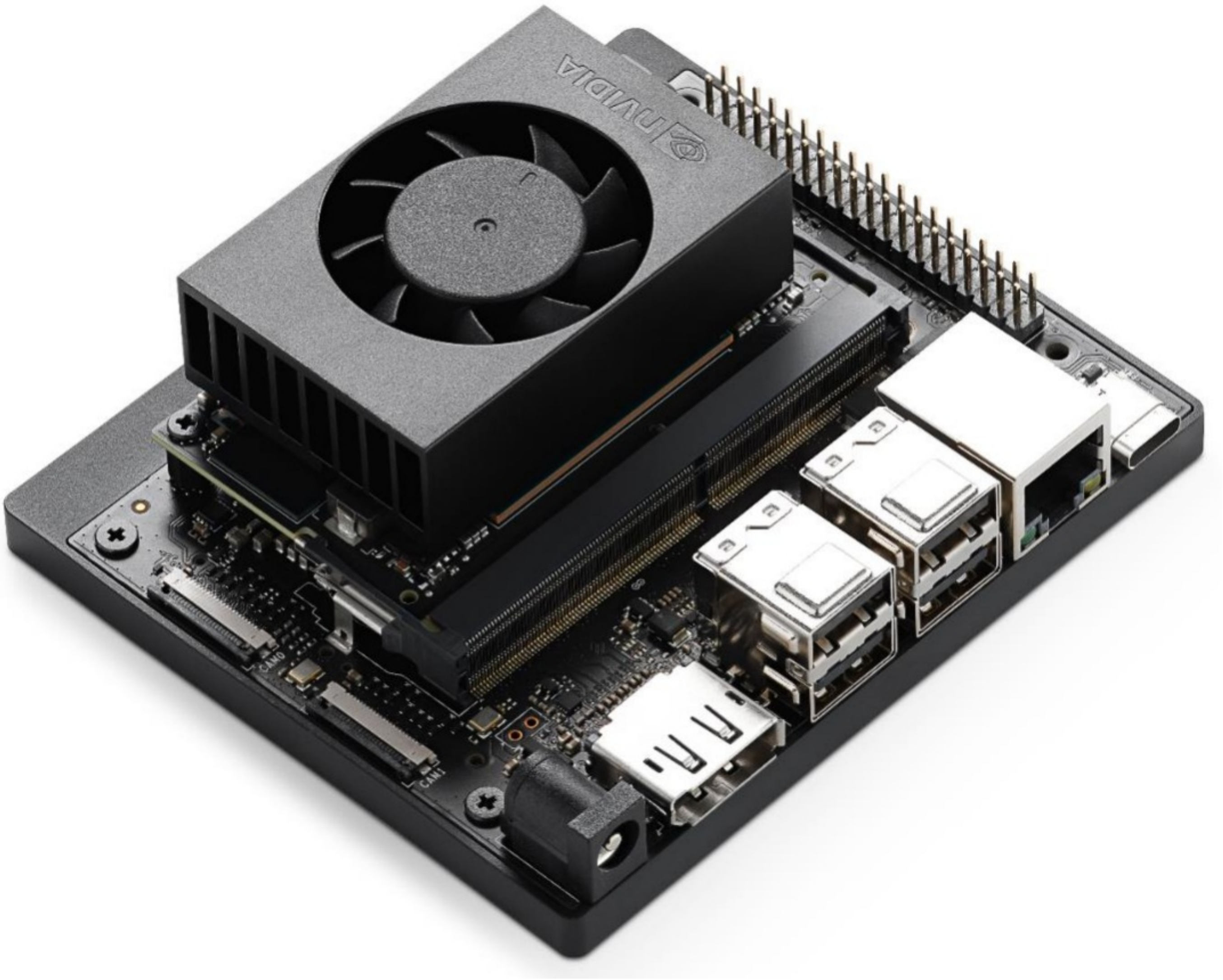


NVIDIA Jetson Orin Nano Developer Kit User Guide

NVIDIA Jetson Orin Nano Developer Kit


NVIDIA
CAM0
CAM1

Contents

Introducing NVIDIA® Jetson Orin™ Nano Developer Kit

The NVIDIA® Jetson Orin™ Nano Developer Kit sets a new standard for creating entry-level AI-powered robots, smart drones, and intelligent cameras, and simplifies getting started with the NVIDIA Jetson platform. The Jetson Orin Nano Developer Kit delivers class-leading performance of up to 40 TOPS in a compact form factor along with a full range of IO connectors making it the perfect developer kit for transforming your visionary concepts into reality. NVIDIA Jetson Orin Nano delivers the following key benefits:

- Up to 80X the AI performance of the previous generation NVIDIA® Jetson Nano™.
- Supports a wide variety of AI models, including transformers and advanced robotics models.
- Faster Time-to-Market with the NVIDIA AI software stack.

The developer kit includes a Jetson Orin Nano 8GB module and a reference carrier board that can support all Jetson Orin Nano and Orin NX modules, providing an ideal platform for prototyping next-gen edge AI products. The Jetson Orin Nano 8GB module features an NVIDIA® Ampere GPU (with 1024 CUDA® cores and 32 third-generation Tensor cores) and a 6-core ARM CPU, capable of running multiple concurrent AI application pipelines and delivering high inference performance. The carrier board included in the developer kit comes with a wide array of connectors, including two MIPI CSI connectors supporting camera modules with up to 4-lanes, allowing higher resolutions and frame rates.

The Jetson Orin Nano Developer Kit is priced at $499 and is available for purchase through NVIDIA authorized distributors worldwide.

Developer kit contents

- Jetson Orin Nano 8GB module with heat sink and reference carrier board
- DC Power Supply
- 802.11ac/abgn wireless network interface controller
- Quick Start Guide

Jetson Orin Nano Developer Kit features

MODULE	**Orin Nano 8GB Module**
GPU	NVIDIA Ampere architecture with 1024 NVIDIA® CUDA® cores and 32 Tensor cores
CPU	6-core Arm Cortex-A78AE v8.2 64-bit CPU 1.5MB L2 + 4MB L3
Memory	8GB 128-bit LPDDR5 68 GB/s
Storage	external via microSD slot external NVMe via M.2 Key M

Table 1 Jetson Orin Nano Developer Kit Module Specs

Power	7W to 15W

Table 1 Jetson Orin Nano Developer Kit Carrier Board Specs

REFERENCE CARRIER BOARD:	
Camera	2x MIPI CSI-2 22-pin Camera Connectors
M.2 Key M	x4 PCIe Gen 3
M.2 Key M	x2 PCIe Gen3
M.2 Key E	PCIe (x1), USB 2.0, UART, I2S, and I2C
USB	Type A: 4x USB 3.2 Gen2 Type C: 1x for Debug and Device Mode
Networking	1x Gbe Connector
Display	DisplayPort 1.2 (+MST)
microSD slot	UHS-1 cards up to SDR104 mode
Others	40-Pin Expansion Header (UART, SPI, I2S, I2C, GPIO) 12-pin button header4-pin fan header DC power jack
Dimensions	100 mm x 79 mmx 21 mm (Height includes feet, carrier board, module, and thermal solution)

Refer to section Setting up your Jetson **Orin Nano Developer Kit** in appendix on how to easily setup the developer kit.

New Performance Standard for Entry-Level AI Applications

Up to 80X Higher AI Performance

The power-efficient Jetson Orin Nano 8GB System-on-Module (SoM) delivers up to 40 INT8 TOPS of AI performance within a 15-Watt power budget, an 80X speedup over the previous generation Jetson Nano1. For applications requiring FP32 precision, Orin Nano 8GB delivers more than 5X the FP32 CUDA TFLOPS of Jetson Nano and with six Arm® A78 CPU Cores, it delivers almost 7X CPU performance. For designs with lower power requirements, the developer kit can be tuned for power profiles as low as 7W. Jetson Orin Nano delivers incredible energy efficiency and is almost 50X more energy efficient than Jetson Nano for AI performance.

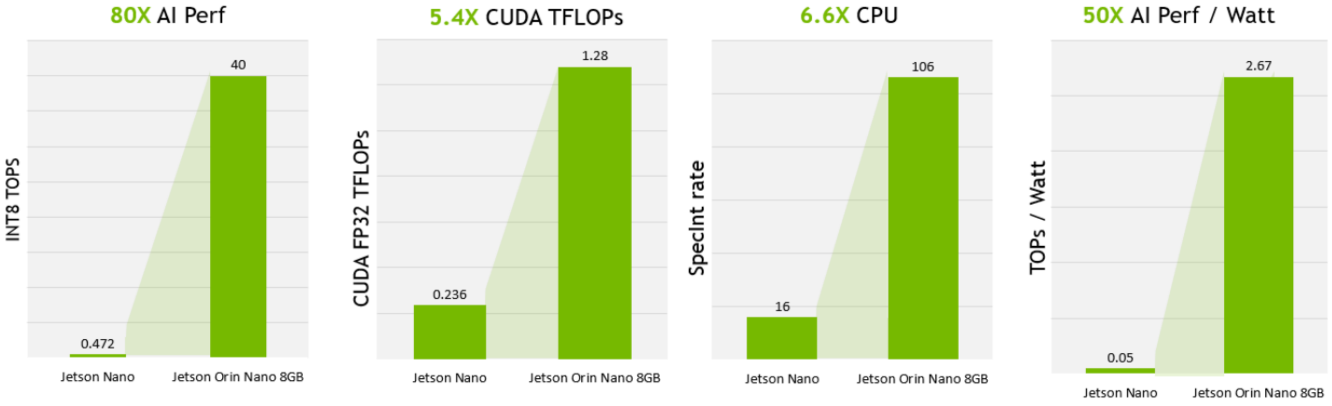


Figure 1 Jetson Orin Nano 8GB delivers 80x the AI performance of Jetson Nano

On the industry standard MLPerf benchmark, Jetson AGX Orin when launched in April of 2022, delivered best in class inference performance that was almost 5X higher than the previous generation Jetson AGX Xavier. Since then, thanks to continual software upgrades to JetPack and the NVIDIA AI stack, AGX Orin's power efficiency has further improved by almost 50 percent. We also have seen up to 54% performance improvements on benchmarks we have run on Jetpack 5.0.2 compared to JetPack 5.1.1 that we will be releasing soon

The same NVIDIA AI architecture that powers the class-leading Jetson AGX Orin module is now accessible to a larger group of developers and AI enthusiasts through the NVIDIA Jetson Orin Nano platform. Jetson Orin Nano delivers tremendous performance on a wide variety of popular AI and computer vision models relative to the previous generation Jetson Nano, enabling developers to create more performant entry-level AI-powered robots, smart drones, and intelligent cameras.

When measured across the popular networks listed below that are widely used in AI and robotics applications, Jetson Orin Nano 8GB on average delivers almost 30X the performance of Jetson Nano, and we expect this lead to improve to almost 45X with continued optimizations and software upgrades in both JetPack and the NVIDIA AI stack.

- **NVIDIA PeopleNet v2.3** for pruned people detection, and NVIDIA PeopleNet v2.5 for the highest accuracy people detection
- NVIDIA ActionRecognitionNet 2D and 3D models
- NVIDIA LPRNet for license plate recognition
- **NVIDIA DashCamNet, BodyPoseNet** for multi-person human pose estimation
- ResNet-50 (224×224) Model

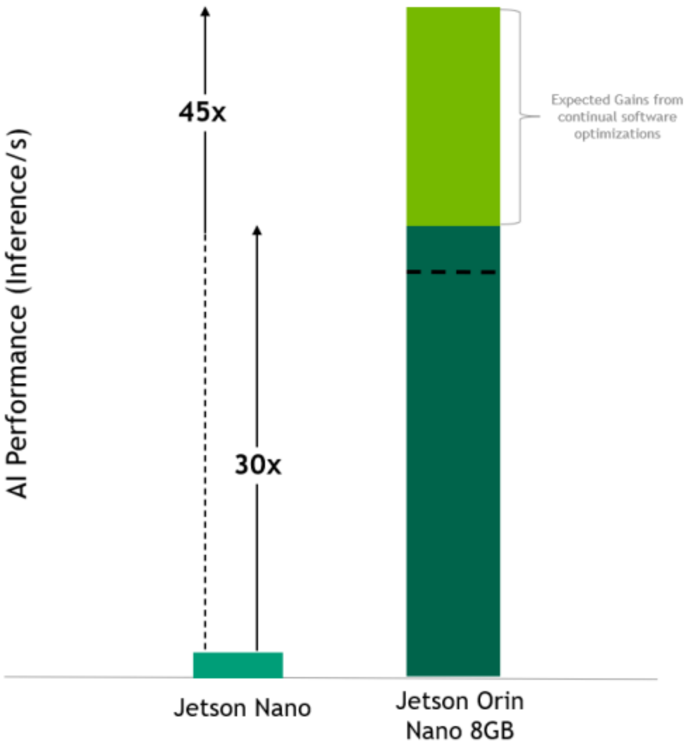
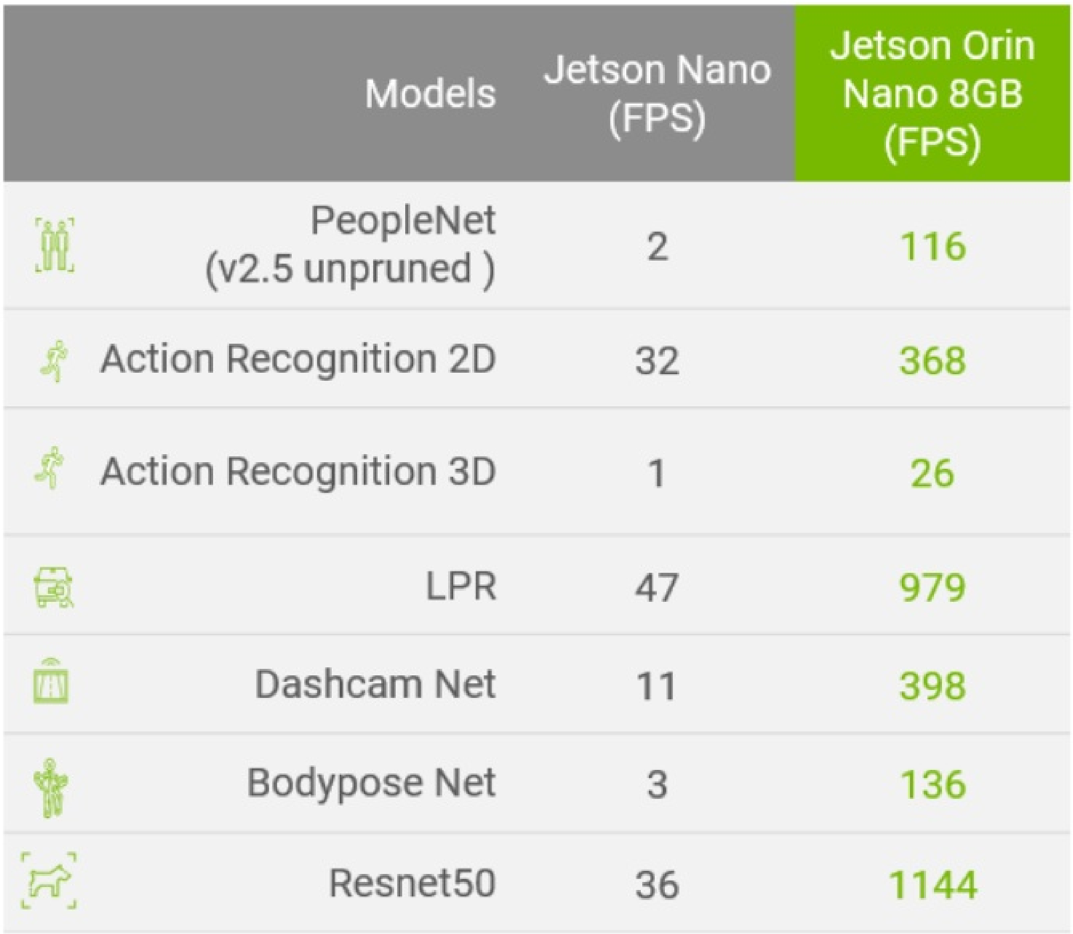


Figure 2 Jetson Orin Nano delivers more than 30x the AI performance of Jetson Nano[2]

Performance on Vision AI and Conversational AI Models



Models	Jetson Nano (FPS)	Jetson Orin Nano 8GB (FPS)
PeopleNet (v2.5 unpruned)	2	116
Action Recognition 2D	32	368
Action Recognition 3D	1	26
LPR	47	979
Dashcam Net	11	398
Bodypose Net	3	136
Resnet50	36	1144

Table 2 Jetson Orin Nano Performance on widely used Vision AI models

2 Relative performance gain represents the geometric mean of performance gains measured across a wide variety of production-ready pre-trained neural networks and inference models.

Instructions for running the above benchmarks on Jetson Orin Nano are available in **Running Inference Benchmarks** section of Appendix.

Runs Cutting-Edge AI Models

Jetson Orin Nano Developer Kit with up to 40 TOPS of AI performance is the most versatile edge AI platform that supports not just the prevalent AI models used in edge AI applications but also newer cutting-edge models such as Transformers that are generating a lot of excitement in the AI industry. Edge devices powered by Jetson Orin Nano can locally run Transformer AI models, the basis for modern generative AI applications such as Chat GPT and DALL-E. This ability to run modern, demanding AI models locally on-device enables developers to build solutions that don’t depend on powerful servers in the datacenter and deploy these solutions for autonomous operations independent of network connectivity.

Transformers are a type of neural network architecture that has been gaining popularity in recent years, especially for natural language processing (NLP) tasks. In the field of computer vision, convolutional neural networks (CNNs) have been the dominant approach for several years now. However, the success of transformer models in NLP has led researchers to recently start exploring the use of transformer models for vision tasks, with promising results.

Transformer models are more robust in handling noisy data, dealing with new previously unseen data, and have been found to deliver better accuracy in situations where traditional CNN model accuracy declines sharply. Transformer models are compute-heavy and require tremendous amounts of compute to train and to deploy on the edge. As research in Transformer-based models for computer vision and other areas continues and new optimized models are created, the versatile and powerful Jetson Orin line of products will be capable of running many of these newer models.

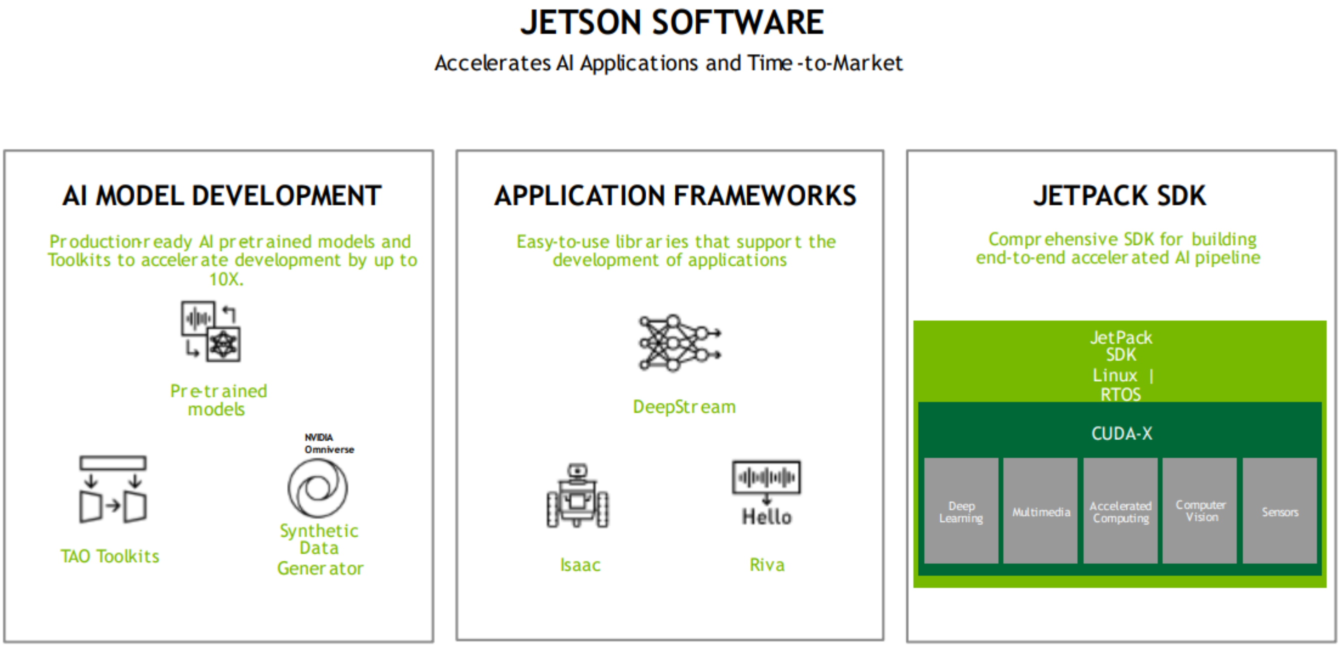
DEMO: Transformer model for people detection
We have packaged a demo for you to run the PeopleNet D-DETR transformer model on your Jetson Orin Nano Developer Kit. NVIDIA offers many transformer-based models on NGC, including this people detection model based on Deformable DETR. DETR (DEtection TRansformer) replaces the traditional region proposal network (RPN) used in CNN-based object detection models with a transformer-based encoder-decoder architecture. The PeopleNet transformer model is based on the Deformable DETR object detector with ResNet50 as a feature extractor. This architecture utilizes attention modules that only attend to a small set of key sampling points around a reference; this optimizes training and inference speed.

Run the PeopleNet D-DETR transformer model by referring to **Peoplenet Transformer Model** section in the Appendix

Accelerates AI Development

The powerful NVIDIA Jetson Orin line up is backed by the same comprehensive NVIDIA AI software stack that powers NVIDIA GPU-based datacenter servers, AI workstations, GeForce gaming computers and the entire Jetson family of products. The NVIDIA AI software stack includes a wide variety of Software Developer Kits (SDK), tools, libraries, pretrained models, and containers that not only accelerate AI applications but also enable a seamless development journey from concept to production deployment. The comprehensive software stack coupled with detailed documentation, tutorials, sample applications, containers, and GitHub repos enables even developers with little to no AI expertise to create production ready AI applications for edge deployments.

The NVIDIA AI software stack provides solutions to accelerate each part of the AI application development journey, starting with tools such as NVIDIA Omniverse Replicator for data generation used in training AI models, NVIDIA TAO that simplifies the actual training and optimization of AI models, and NVIDIA TensorRT for AI model deployment. Domain-specific SDKs such as NVIDIA DeepStream, NVIDIA Riva, NVIDIA Isaac, and others help in development of end-to-end workflows and applications. NVIDIA continually invests in its AI software stack to bring new capabilities for developers.


JETSON SOFTWARE
Accelerates AI Applications and Time-to-Market
AI MODEL DEVELOPMENT
Production-ready AI pretrained models and Toolkits to accelerate development by up to 10X.
Pre-trained models
TAO Toolkits
NVIDIA Omniverse
Synthetic Data Generator
APPLICATION FRAMEWORKS
Easy-to-use libraries that support the development of applications
DeepStream
Isaac
Hello
Riva
JETPACK SDK
Comprehensive SDK for building end-to-end accelerated AI pipeline
JetPack SDK
Linux | RTOS
CUDA-X
Deep Learning
Multimedia
Accelerated Computing
Computer Vision
Sensors

AI Model Development

AI model accuracy greatly depends on the amount and quality of the data used for training. Collecting a large amount of data in different scenarios and then labeling them for training is an arduous job and slows time to market. **NVIDIA Omniverse Replicator** for synthetic data generation helps create high- quality datasets to boost model training. Datasets created with or augmented by synthetic data seamlessly work with NVIDIA **Train-Adapt-Optimize (TAO) Toolkit** to quickly train and optimize a custom model or one of the many **Pre-Trained Models** (PTM) hosted on NGC™ (NVIDIA GPU Cloud).

The recent NVIDIA TAO release added support for AutoML that enables developers to easily train AI models without going through the hassle of manually fine-tuning hundreds of parameters, thus significantly reducing the time required to optimize a model. The latest release also enabled developers to bring their own model and convert any open-source ONNX (Open Neural Network Exchange) model to a TAO-compatible model. This release also enables developers to access TAO toolkit services through REST APIs making it easier for them to integrate TAO into their AI model lifecycle.

Powerful Software Developer Kits

NVIDIA AI software stack comes with a variety of SDKs that help developers easily create fully accelerated AI applications. Three of the key SDKs that help in the development of vision, robotics, and conversational AI are highlighted below.

NVIDIA DeepStream SDK provides a framework based on gstreamer for building applications that can analyze multiple video streams in real-time. DeepStream offers hardware acceleration for more than just inference — it includes hardware-accelerated plugins for end-to-end AI pipeline acceleration. The SDK now includes support for REST APIs for controlling DeepStream pipelines on the fly, making it easy to integrate with your applications. An updated Graph Composer with an easy-to-use UI lets you create end-to-end DeepStream pipelines with little to no code, thereby reducing the complexity of application development and providing another reduction in time to market.

NVIDIA Riva includes **state-of-the-art pre-trained models** for Automatic Speech Recognition (ASR) and Text-To-Speech (TTS). These pre-trained models are highly accurate and can be easily customized to improve accuracy on desired domains, accents, languages, and use cases. The recent release of Rivanow supports 14 languages with plans to continue adding more. We added support for Fast Conformer that delivers 2.4x inference speedup and 1.8x training speedup over the standard Conformer. We have also made our models more noise robust leading to >20% improvement in accuracy on noisy test sets. The latest release also supports Translation skill. We have added high quality out-of-the-box translation models with support for 32 languages.

End-to-End Robotics Deployment with NVIDIA Isaac

NVIDIA Isaac provides a platform for end-to-end robotics development. Start with synthetic data generation using **Omniverse Replicator.** Developers can easily create realistic scenes with different people, objects, and environments, randomize lighting conditions, colors and object positions, and output a large, perfectly labeled dataset to accelerate model training. Use pre-trained models from NVIDIA NGC instead of developing models from scratch and use TAO to train and optimize the model.

Then use Isaac Sim to simulate your robot physically accurately in photorealistic environments and create digital twins. You can test each aspect of the robot operation – may it be perception, localization, path planning, navigation, and so on. You can simulate and test conditions which may not be even possible to re-create in the real world. The 2022.2 release of Isaac Sim brought support to simulate people and actions, so that you can validate perception and safety systems in a virtual world.

The release also brought support for cuOpt to optimize the path planning of robots in warehouse and factory environments. In addition, the latest release added support for ROS 2 Humble so you can simulate your Isaac ROS code within Isaac Sim.

Next, you can build high performance robotic applications using **NVIDIA Isaac ROS,** a collection of accelerated packages that robotics developers can easily integrate in their ROS pipelines. These ROS packages are accelerated on GPU and other hardware accelerators available on Jetson. With the latest release (targeted for GTC March 2023), we are open-sourcing the Isaac ROS packages for ROS developers to modify, extend, or contribute back. We are also releasing a ROS benchmarking framework to create benchmarks for any ROS graph and benchmark under realistic workloads. An updated Isaac ROS NVBlox package is being released to generate a clean 3D mesh in a crowded environment. With these and other enhancements, Isaac ROS is laying the foundation for high performance robotic solution creation.

DEMO: End-to-End Workflow with Isaac Sim Synthetic Data Generation and TAO

This demo provides you with the experience of going through an entire workflow from model creation to deployment. We start with the pre-trained People Segmentation model from NGC, which was trained with a dataset consisting of various camera heights, crowd-densities, and fields-of view (FOV). We observe that using this model on a close-to-ground robot camera results in poor accuracy because the original training dataset did not have such close-to-ground camera angles.

Collecting a new, large dataset with the new camera angle and then annotating each image would take many days and weeks. Instead, this demo shows the use of Omniverse Replicator in Isaac Sim to create scenes, objects, and people, and to add randomization for more variations in data. The resulting synthetic data is already annotated. This synthetic data is then used to retrain the people segmentation model before optimizing it with NVIDIA TAO toolkit. We then deploy the trained model on the Jetson Orin Nano Developer Kit and see the inferencing results: the model will accurately perform with the close-to-ground robot camera angle.

With tools like Isaac sim for synthetic data generation, pre-trained models, and NVIDIA TAO, we enable our customers get to market faster by reducing AI model development time from days and weeks to only minutes.

Experience this entire workflow by referring to **Isaac Replicator and TAO** section in the Appendix

Appendix

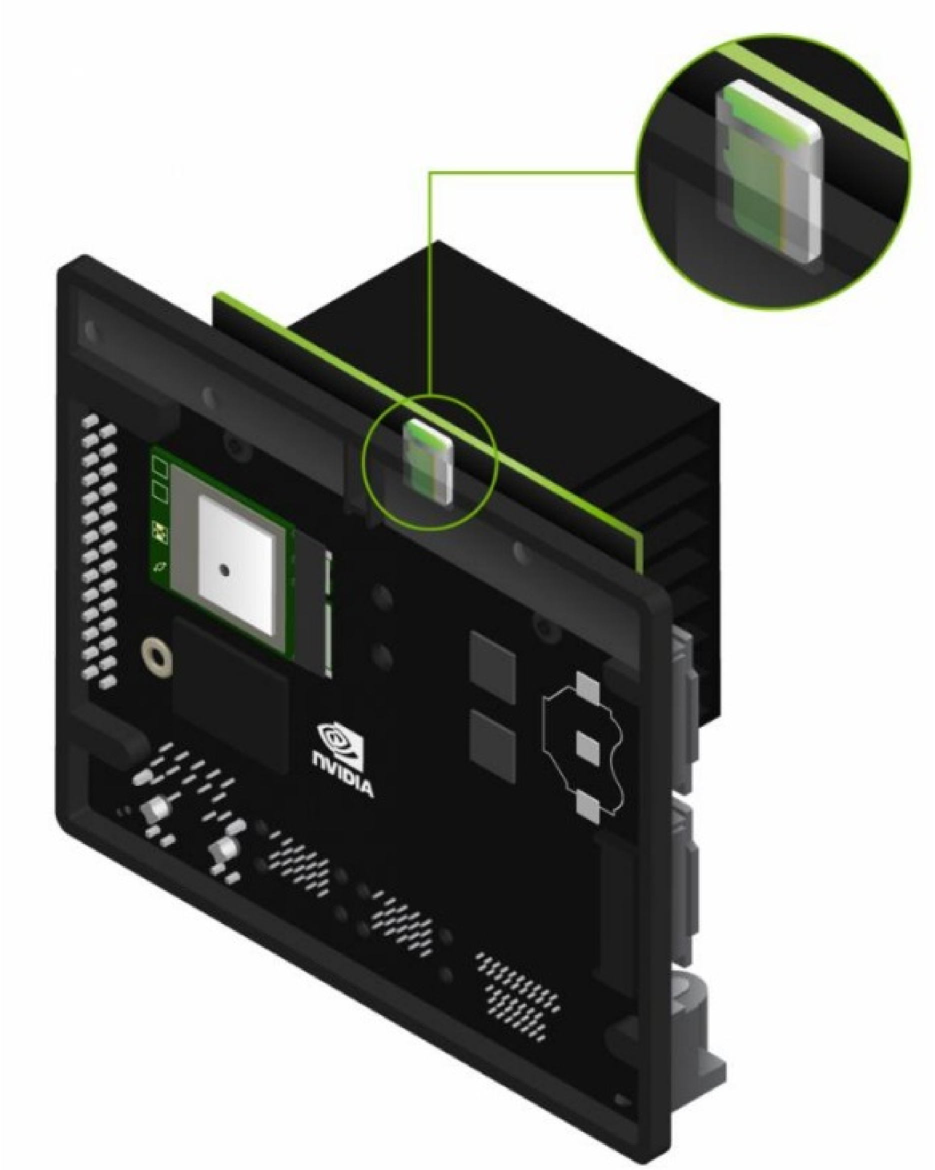
Getting Started with the Jetson Orin Nano Developer Kit

You can access this Reviewers Guide from **https://developer.nvidia.com/jetson-orin-nano-review**

Setting up your Jetson Orin Nano Developer Kit

The Jetson Orin Nano Developer Kit comes with a Jetson Orin Nano module which has a microSD Card slot. The easiest way to set up the developer kit is to write the NVIDIA JetPack image to the microSD Card using a tool like Balena Etcher and then boot up the developer kit with the SD card image.

For this review, we have provided a microSD Card which is pre-written with the NVIDIA JetPack image. Please simply insert the SD card into the microSD card slot and power on the developer kit. The microSD Card slot is under the module as shown below.



NOTE: The image we have provided is a **"private"** preview build. This preview build will take a minute or so to show the initial configuration dialogue during first boot. The developer kits, which will be made available for public after announcement at GTC 2023, will have a production quality build which will be much faster at the first boot. Also, there is an issue in this private build that needs attention. Jetson Orin Nano does not include a hardware encoder. Please refer to this section on using a software encoder on Jetson. In this preview build, any attempt to use a hardware encoder will result in system freeze. This issue will be fixed in the production release. Also you will see that the available memory will be shown as **6.3GB** in this preview build instead of 8GB. This is due to some memory carved out for security purposes. NVIDIA will be fixing this issue and automatically reclaiming memory carveouts which are not in use.

The first boot on this preview image will guide you through the initial configuration dialogue. Go through the simple initial setup process. Once the initial configuration is complete and the developer kit has booted to the desktop, you can start exploring the developer kit.

The developer kit is running JetPack 5.1.1 (pre-release) which has following components:

- CUDA 11.4
- TensorRT 8.5
- cuDNN 8.6
- VPI 2.2

- Vulkan 1.3
- Nsight Systems 2022.5
- Nsight Graphics 2022.6

On top right of the desktop, there is a power profile selector. When clicked, it provides a drop-down menu of all software defined power modes for the developer kit. JetPack comes with **multiple samples built in.**
These samples provide a preview and capabilities of different JetPack components.

Running Inference Benchmarks

Download the tar ball named “benchmarking.tar.gz” required for this benchmarking from **here** and untar the tarbal

```
$ tar -xvf benchmarking.tar.gz
```

Set up the requirements for running the benchmark by doing:

```
$ cd benchmarking
$ sudo bash install_requirements.sh
```

For a clean measurement, please reboot the system and then start benchmarking.

Run Vision Model Benchmarks using:

```
$ sudo python3 benchmark.py –all –csv_file_path orin_nano_ptm.csv \ –model_dir <absolute-path-to-model-egines-directory>
```

Example:

```
$ sudo python3 benchmark.py –all –csv_file_path orin_nano_ptm.csv \ –model_dir /home/nvidia/benchmarking/model_engines
```

You will get the output similar to below at the end of benchmarking. Please note that benchmarking will take several minutes to complete.

Model	Name	FPS
0	peoplenet	110.102016
1	action_recog_2d	382.461916
2	action_recog_3d	25.946016
3	dashcamnet	395.563297
4	bodyposen et	135.222464
5	lpr_us	981.867680

Run PeopleNet Transformer Model

We have created a DeepStream container with PeopleNet transformer model and hosted it on NGC

Log in to NGC

```
$ sudo docker login nvcr.io
```

Give username as “$oauthtoken” and password as “MnBxcWF2ZWtibGRzNm1yZmx2c3R0ZWx2NGk6NjA1MmM1NjgtZGQxNi00YmJjLWFmNzktOGM5NDg2ODFlMGRj”

Like this

Username: $oauthtoken

Password: MnBxcWF2ZWtibGRzNm1yZmx2c3R0ZWx2NGk6NjA1MmM1NjgtZGQxNi00YmJjLWFmNzktOGM5NDg2ODFlMGRj

Pull the container

```
$ sudo docker pull nvcr.io/ea-linux4tegra/deepstream-reviewers:latest
```

Pull the Directory

Download the tar ball named mount_dir from **here** and extract it in your home directory. This directory contains video files which will act as input streams for inferencing.

```
$ tar -xvf mount_dir.tar.gz -C ${HOME}/
```

After extraction, you should have a folder with the name “mount_dir” in your home directory.

Run the container

```
$ xhost +
$ sudo docker run -it –rm –name=ds_docker –net=host –runtime nvidia -e \
DISPLAY=$DISPLAY -w /opt/nvidia/deepstream/deepstream-6.2 \
-v /tmp/.X11-unix/:/tmp/.X11-unix -v $HOME/mount_dir:/mount_dir \
nvcr.io/ea-linux4tegra/deepstream-reviewers:latest
```

Run the Model

Run the following command inside the container to start inferencing

```
$ cd /opt/nvidia/deepstream/deepstream-6.2/samples/configs/deepstream-app-triton
$ deepstream-app -c source1_primary_detector_peoplenet_transformer.txt
```

Please note that the above DeepStream config file has been modified to look for the input stream in the mount_dir.

This PeopleNet D-DETR transformer model currently is a FP16 model and runs at 8 FPS on Jetson Orin Nano which is useful for many applications not requiring real time performance. The model runs at around 30 FPS on Jetson AGX Orin. As the research continues on transformer models for Vision, these models will be optimized further for performance at the edge.

NVIDIA Omniverse Replicator and TAO

We have created internal cloud instances for you to run Isaac Sim and TAO. Please reach out to us at **JONReviewersTeam@nvidia.com** for your cloud instance credentials. To access this internal instance, it is required to install the VMWare Horizon Client. Since there is no Arm version of the VMWare client, we request you to use your own laptop or PC to access the internal cloud instance and run through the workflow we have for you.

Synthetic Data Generation and Training the Model
You can use any laptop or PC (Windows, Linux, Mac or Chromebook). Please follow the instructions by referring to document named “VMWare Horizon Client to Access Cloud Instance” from **here** to install VMWare Horizon client and access the internal cloud instance and run through the workflow. The workflow will guide you through generating synthetic data using Omniverse Replicator, then training and optimizing people segmentation pre-trained model from NGC on synthetic data, and then downloading the trained model. Once the model is downloaded to your PC, please copy the model to Jetson either by doing “scp” or by using a storage media like a USB stick.

Inferencing on Jetson

Please make sure the model that you trained in the cloud and copied to Jetson is named “shuffleseg_exported.etlt”.
Download the tar ball named mount_dir from **here** and extract it in your home directory (if you had already downloaded the mount_dir for other demo, you can skip this).

```
$ tar -xvf mount_dir.tar.gz -C ${HOME}/
```

Copy the model into this directory (if there was already an existing one, please replace it with your trained model). Please note that the below DeepStream app will look for the input stream and the model in the mount_dir. Change the model to have read permission if not already by following command like this:

```
$ chmod +r shuffleseg_exported.etlt
```

Next pull the DeepStream container by following the instructions below. If you had already pulled the container for the previous demo, you can skip this step

Log in to NGC

```
$ sudo docker login nvcr.io
```

Give username as “$oauthtoken” and password as “MnBxcWF2ZWtibGRzNm1yZmx2c3R0ZWx2NGk6NjA1MmM1NjgtZGQxNi00YmJjLWFmNzktOGM5NDg2ODFlMGRj”

Like this

Username: $oauthtoken

Password: MnBxcWF2ZWtibGRzNm1yZmx2c3R0ZWx2NGk6NjA1MmM1NjgtZGQxNi00YmJjLWFmNzktOGM5NDg2ODFlMGRj

Pull the container

```
$ sudo docker pull nvcr.io/ea-linux4tegra/deepstream-reviewers:latest
```

Run the container

```
$ xhost +
$ sudo docker run -it –rm –name=ds_docker –net=host –runtime nvidia -e \
DISPLAY=$DISPLAY -w /opt/nvidia/deepstream/deepstream-6.2 \
-v /tmp/.X11-unix/:/tmp/.X11-unix -v $HOME/mount_dir:/mount_dir \
nvcr.io/ea-linux4tegra/deepstream-reviewers:latest
```

Run the DeepStream Inferencing

Inside the container run following commands.

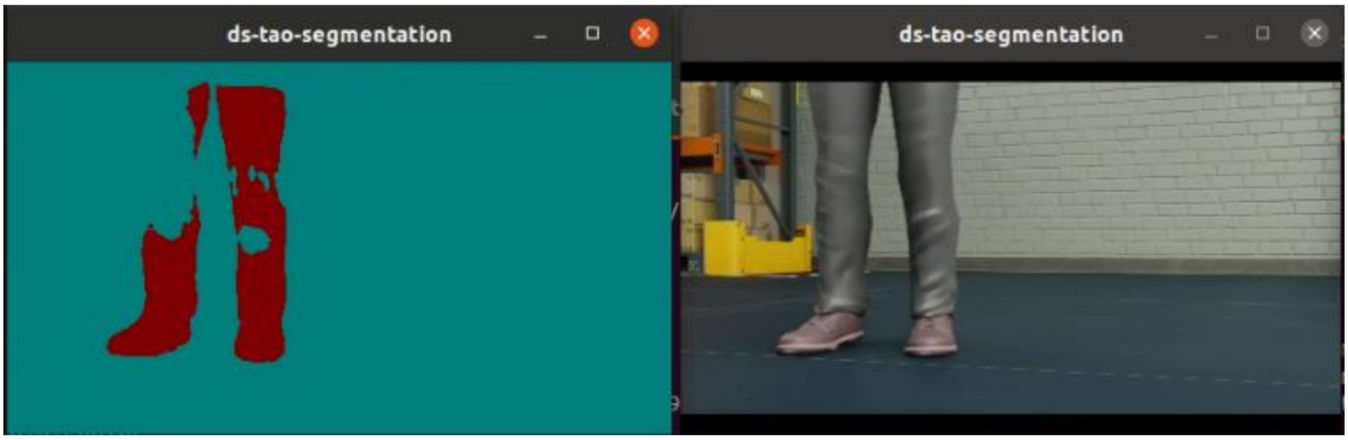
Original Model

We will first run the inferencing on the original model from NGC, which was not pre-trained with synthetic data

```
$ cd /opt/nvidia/deepstream_tao_apps

$ ./apps/tao_segmentation/ds-tao-segmentation -e -d \
-c configs/peopleSemSegNet_tao/shuffle/pgie_peopleSemSegShuffleUnet_tao_config.txt \
-i file:///mount_dir/carter_lower_perspective.mp4
```

You will see that two windows will pop up. One showing the input video and the other showing People segmentation output as shown below. You can see how the model does not perform accurately.

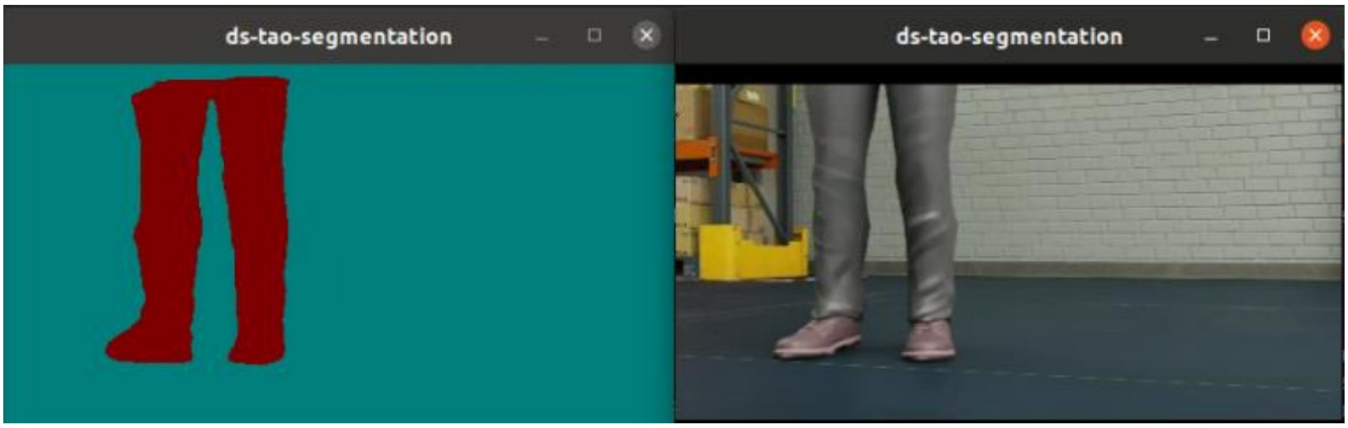


<u>Model trained with synthetic data</u>

Next run inferencing on the model you trained with synthetic data.

```
$ cd /opt/nvidia/deepstream_tao_apps
$ ./apps/tao_segmentation/ds-tao-segmentation -e -d -c \
configs/peopleSemSegNet_tao/sdg_shuffle/pgie_unet_tlt_config_peoplesemsegnet_shuffleseg.txt \
-i file:///mount_dir/carter_lower_perspective.mp4
```

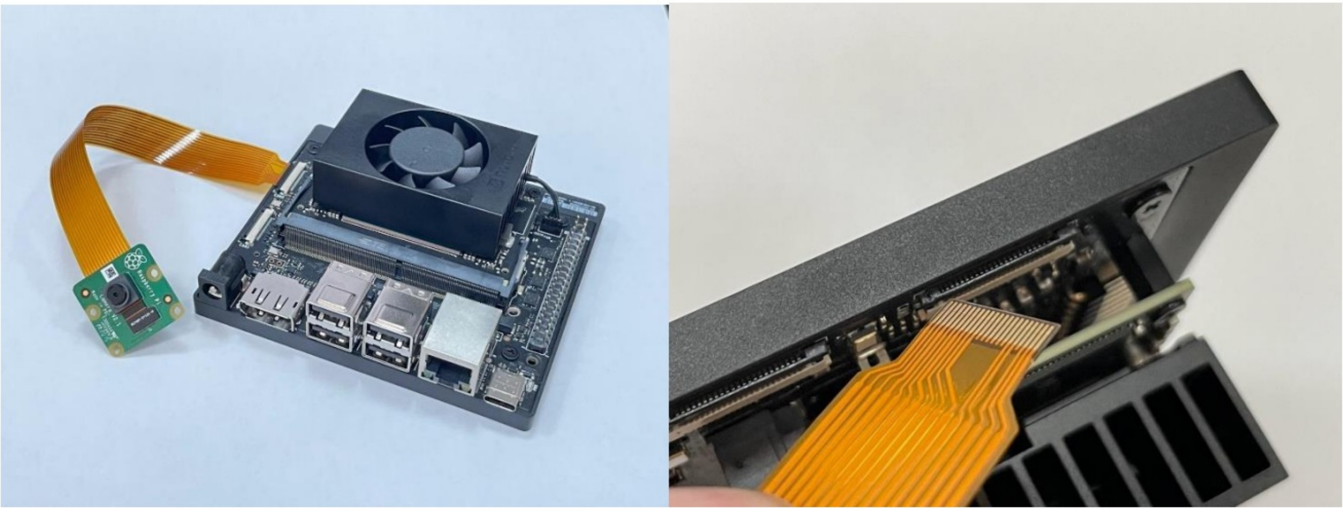
You will see that two windows will pop up. One showing the input video and the other showing People segmentation output like shown below. Compared to the inferencing output when running the original model, you can see that the model trained with synthetic data is performing more accurately. For the purpose of the review, we trained with limited synthetic dataset to cut down training time wait for you.
The model can be improved for accuracy when trained with a larger synthetic dataset.



Additional Resources

Connecting CSI Cameras

There are two MIPI CSI-2 4-lane ports on the Jetson Orin Nano Developer Kit carrier board. You will need a 22-pin to 15-pin connector if the camera has a 15-pin connector. Please check the imagesbelow carefully on the orientation of the cable when connecting to the camera ports.



Encoding on Jetson Orin Nano

Jetson Orin Nano does not have an NVIDIA hardware encoder (NVENC) but we have a 6-core ARM A78AE CPU with which you can run software encoding. You can use x264 enc to run SW encoding in your pipelines.

The following command demonstrates the H.264 SW encode using the x264enc plugin with input from the camera plugin that uses Argus API.

```
$ gst-launch-1.0 nvarguscamerasrc ! \
‘video/x-raw(memory:NVMM), width=(int)1920, height=(int)1080, \
format=(string)NV12, framerate=(fraction)30/1’ ! nvvidconv ! \
video/x-raw, format=I420 ! x264enc ! \
h264parse ! qtmux ! filesink \
location=<filename_h264.mp4> -e
```

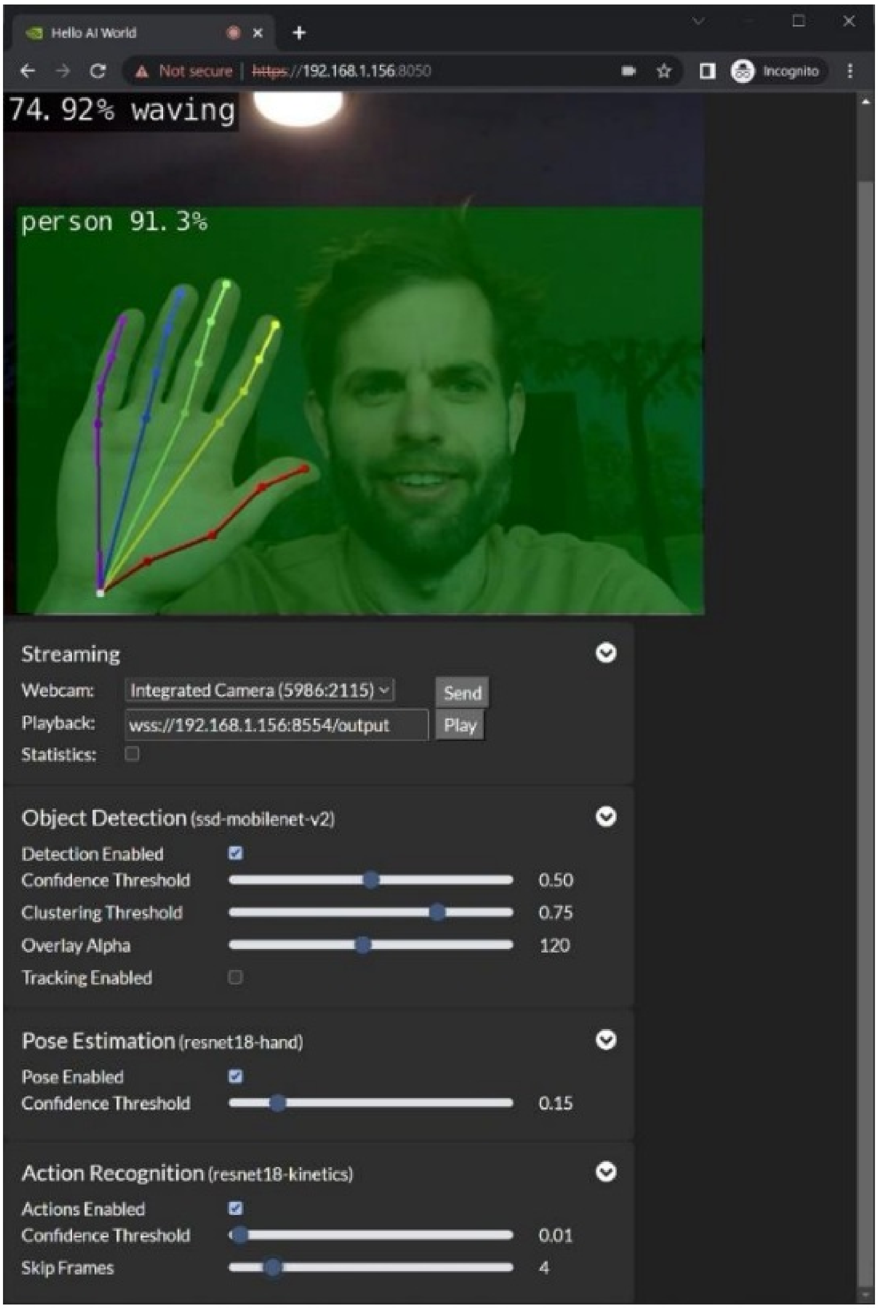
If you are using DeepStream then you can change the encoder-type to use software encoding. For example, see highlighted below

```
[sink0]
#source0 output as filesink
enable=1
#1=h264 2=h265
codec=1
#encoder type 0=Hardware 1=Software
enc-type=1
…
```

WebRTC WebApp Framework

Hello AI World is an open-source deep learning tutorial and library for getting started with training and deploying DNNs on Jetson. It uses TensorRT for real time inference and has easy-to-use Python/C++ APIs for classification, detection, segmentation, pose estimation, and action recognition, along with support for various camera interfaces and video devices.

New to Hello AI World is low-latency WebRTC live video streaming to/from web browsers, along with several examples of integration with popular Python-based web frameworks, including Flask, Plotly Dash, and HTML5/JavaScript. This enables developers to quickly create their own interactive webapps and remote data visualization tools powered by Jetson and edge AI on the backend.



Streaming
Webcam: Integrated Camera (5986:2115)
Playback: wss://192.168.1.156:8554/output
Statistics: □

Object Detection (ssd-mobilenet-v2)
Detection Enabled ☑
Confidence Threshold 0.50
Clustering Threshold 0.75
Overlay Alpha 120
Tracking Enabled □

Pose Estimation (resnet18-hand)
Pose Enabled ☑
Confidence Threshold 0.15

Action Recognition (resnet18-kinetics)
Actions Enabled ☑
Confidence Threshold 0.01
Skip Frames 4

To get started with Hello AI World on your Jetson Orin Nano Developer Kit, see the following links:

- **https://github.com/dusty-nv/jetson-inference**
- **Running the Docker Container**
- Tutorial on **WebApp Frameworks**

NVIDIA Contact Information

Any questions when reviewing the Developer Kit? Email **JONReviewersTeam@nvidia.com**

NVIDIA North/Latin America Public Relations

David Pinto
PR Manager, Autonomous Machines
Office: 408 566 6950
dpinto@nvidia.com

Michael Lim
Director, Analyst Relations
Office: 408 486 2376
mlim@nvidia.com

Sridhar Ramaswamy
Senior Director, Enterprise Technical Marketing
and Reviews
Cell: 510 545 3774
sramaswamy@nvidia.com

NVIDIA Europe Public Relations

Jens Neuschäfer Senior PR Manager, Enterprise Europe Office: +49 89 6283 50015 Mobile: +49 173 6282912 **jneuschafer@nvidia.com** NVIDIA GmbH Haus 1 West, 3rd Floor Flössergasse 2 81369 Munich, Germany	**Rick Napier** Senior Technical Product Manager Northern Europe Office: +44 (118) 9184378 Mobile: +44 (7917) 630172 **rnapier@nvidia.com** NVIDIA UK 100 Brook Drive Green Park Reading RG2 6UJ

NVIDIA Asia/Pacific Public Relations

Jeff Yen
Director, Technical Marketing, APAC

Office : +886 987 263 193
jyen@nvidia.com
NVIDIA
8, Kee Hu Road, Neihu Taipei 114
TAIWAN

Melody Tu
PR Director, APAC

Office: +65 93551454
metu@nvidia.com
NVIDIA Singapore Development Pte Ltd
#07-03 Suntec Tower Three, 8 Temasek Blvd, Singapore 038988

Searching Shi
Sr. Technical Marketing Manager, China

Office: +86 75586919016
Email: seshi@nvidia.com

5F BLOCK 8 VISEEN BUSINESSPARK 9 HIGH- TECH 9TH SOUTH ROAD SHENZHEN HI-TECH IND. PARK

SHENZHEN, GUANGDONG Shenzhen 518057 China

Alex Liu
PR/Marketing Manager, China

Office: +86 1058661510
Email: alliu@nvidia.com Fortune Financial Center Level 40, Units 05-2,06
Building #5, Middle Road, East 3rd Ring Chaoyang District, Beijing, China 100000

Kyle Kim
Sr. Technical Marketing Manager, Korea

Office: +82 2 6001 7186

kylek@nvidia.com
NVIDIA Korea
#2101, COEX Trade Tower, 159-1
Samsung-dong, Kangnam-gu, Seoul 135-729 KOREA

Sunny Lee
Marketing Director, Korea

Office: +82 2 6001 7123
slee@nvidia.com
NVIDIA Korea
#2101, COEX Trade Tower, 159-1
Samsung-dong Kangnam-gu, Seoul 135-729 KOREA

Kaori Nakamura Head of Public Relations, Japan Office : +81 3 6743 8712 knakamura@nvidia.com ATT New Tower 13F 2-11-7 Akasaka,Minato-ku, Tokyo 107-0052 , JAPAN	**Masaki Sawai** Technical Marketing Manager, Japan Office: +81 3 6743 8717 Email: msawai@nvidia.com ATT New Tower 13F 2-11-7 Akasaka,Minato- ku, Tokyo 107-0052, JAPAN
John Gillooly Technical Marketing Manager, Asia Pacific South Office: +65 8286 8727 Email: jgillooly@nvidia.com SINGAPORE	**Titus Su** Technical Marketing Manager, TASA Office: +886 2 6605 5430 Email: tisu@nvidia.com 8, Kee Hu Road, Neihu Taipei 114, TAIWAN

Notice

ALL INFORMATION PROVIDED IN THIS REVIEWER'S GUIDE, INCLUDING COMMENTARY, OPINION, NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, the NVIDIA logo, GeForce, Tegra, and Jetson are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. All rights reserved. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

Documents / Resources

 NVIDIA® JETSON ORIN™ NANO DEVELOPER KIT Accelerating Entry-Level Edge AI and Robotics Applications Reviewer's Guide 1	**NVIDIA Jetson Orin Nano Developer Kit** [pdf] User Guide Jetson Orin Nano Developer Kit, Jetson Orin Nano, Developer Kit, Orin Nano Developer Kit, Nano Developer Kit

References

- Hopper Sweeps AI Inference Tests in MLPerf Debut | NVIDIA Blogs
- Action Recognition Net | NVIDIA NGC
- BodyPoseNet | NVIDIA NGC
- DashCamNet | NVIDIA NGC
- License Plate Recognition | NVIDIA NGC
- PeopleNet | NVIDIA NGC
- NVIDIA Omniverse Replicator Generates Synthetic Training Data for Robots | NVIDIA Technical Blog
- DeepStream SDK | NVIDIA Developer
- Isaac Robot Operating System (ROS) | NVIDIA Developer
- Jetson Orin Nano Developer Kit Review | NVIDIA Developer
- Riva Speech AI SDK - Get Started | NVIDIA Developer
- Introduction to JetPack :: NVIDIA JetPack Documentation
- GitHub - dusty-nv/jetson-inference: Hello AI World guide to deploying deep-learning inference networks and deep vision primitives with TensorRT and NVIDIA Jetson.
- GitHub - dusty-nv/jetson-inference: Hello AI World guide to deploying deep-learning inference networks and deep vision primitives with TensorRT and NVIDIA Jetson.
- jetson-inference/aux-docker.md at master · dusty-nv/jetson-inference · GitHub
- GitHub - NVIDIA-AI-IOT/jetson_benchmarks: Jetson Benchmark
- ISAAC Platform For Robotics | NVIDIA