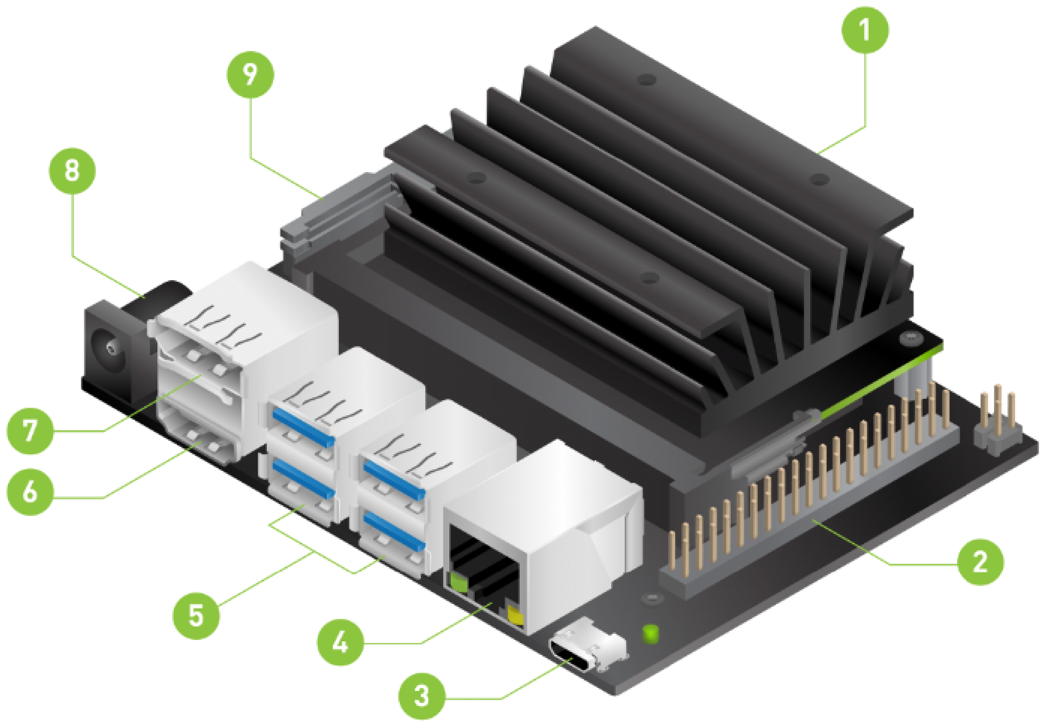
Introduction to Jetson Nano

1.Introduction to the official version of BO1

The physical image of the Jetson NanoBO1 public version is shown in the following figure. One of them is the TF card interface, which can perform system image burning and writing; 2 is a 40PIN GPIO extension interface; 3 is a Micro USB interface used for data transmission or power supply; 4 is a Gigabit Ethernet port; 5 is the USB3.0 interface; 6 is an HDMI interface; 7 is the Display Port interface used to connect DP screens; 8 is the DC power interface; 9 is the interface for connecting the camera; 10 is the Poe interface.



The Jetson Nano development kit covers an area of only 80x100mm and features four high-speed USB 3.0 ports, MIPI CSI-2 camera connectors, HDMI 2.0 and DisplayPort 1.3, Gigabit Ethernet, M.2 Key-E module, MicroSD card slot, and 40 pin GPIO connector. Ports and GPIO connectors are out of the box, featuring various popular peripheral devices, sensors, and out of the box projects, such as NVIDIA's 3D printable deep learning JetBot open-source on GitHub.

The public version can be started from a mobile MicroSD card, or by inserting a USB drive into a USB port. The system can be formatted and imaged from any PC with an SD card adapter through a USB drive. It can be conveniently powered through a Micro USB port or a 5V DC barrel jack adapter. The camera connector is compatible with affordable MIPI CSI sensors, including modules based on the 8MP IMX219 provided by JPS ecosystem partners. It also supports Raspberry Pi Camera Module v2, which includes driver support in JetPack. The following figure shows the technical specifications

处理	
中央处理器	64位四核ARM A57 @ 1.43GHz
GPU	128核NVIDIA Maxwell @ 921MHz
记忆	4GB 64位LPDDR4 @ 1600MHz | 25.6 GB / s
视频编码器*	4Kp30 | （4x）1080p30 | （2x）1080p60
视频解码器*	4Kp60 | （2x）4Kp30 | （8x）1080p30 | （4x）1080p60
接口	
USB	4x USB 3.0 A（主机）| USB 2.0 Micro B（设备）
相机	MIPI CSI-2 x2（15位Flex连接器）
显示	HDMI | DisplayPort的
联网	千兆以太网（RJ45）
无线	M.2带有PCIe x1的Key-E
存储	MicroSD卡（建议最低16GB UHS-1）
其他I / O.	（3x）I2C | （2x）SPI | UART | I2S | 个GPIO

2.Public version deep inference benchmark

Jetson Nano can run a variety of advanced networks, including full native versions of popular ML frameworks such as TensorFlow, PyTorch, Caffe/Cafe2, Keras, MXNet, and more. By implementing powerful functions such as image recognition, object detection and localization, pose estimation, semantic segmentation, video enhancement, and intelligent analysis, these networks can be used to build automated machines and complex AI systems

The following figure shows the inference benchmark test results of popular models provided online. Including performance from other platforms such as Raspberry Pi 3, Intel Neural Compute Stick 2, and Google Edge TPU Coral Dev Board:

模型	应用	骨架	NVIDIA Jetson Nano	覆盆子3	Raspberry Pi 3 + Intel神经计算棒2	Google Edge TPU 开发板
ResNet-50（224×224）	分类	TensorFlow	36 FPS	1.4 FPS	16 FPS	DNR
MobileNet-v2（300×300）	分类	TensorFlow	64 FPS	2.5 FPS	30 FPS	130 FPS
SSD ResNet-18（960×544）	物体检测	TensorFlow	5 FPS	DNR	DNR	DNR
SSD ResNet-18（480×272）	物体检测	TensorFlow	16 FPS	DNR	DNR	DNR
SSD ResNet-18（300×300）	物体检测	TensorFlow	18 FPS	DNR	DNR	DNR
SSD Mobilenet-V2（960×544）	物体检测	TensorFlow	8 FPS	DNR	1.8 FPS	DNR
SSD Mobilenet-V2（480×272）	物体检测	TensorFlow	27 FPS	DNR	7 FPS	DNR
SSD Mobilenet-V2（300×300）	物体检测	TensorFlow	39 FPS	1 FPS	11 FPS	48 FPS
成立V4（299×299）	分类	PyTorch	11 FPS	DNR	DNR	9 FPS
小小的YOLO V3（416×416）	物体检测	暗网	25 FPS	0.5 FPS	DNR	DNR
OpenPose（256×256）	姿势估计	咖啡	14 FPS	DNR	5 FPS	DNR
VGG-19（224×224）	分类	MXNet	10 FPS	0.5 FPS	5 FPS	DNR
超高分辨率（481×321）	图像处理	PyTorch	15 FPS	DNR	0.6 FPS	DNR
UNET（1x512x512）	分割	咖啡	18 FPS	DNR	5 FPS	DNR

Due to limited memory capacity, unsupported network layer, or hardware/software limitations, DNR (not running) results occur frequently. Fixed function neural network accelerators typically support a relatively narrow set of use cases, with hardware supporting dedicated layer operations and requiring network weights and activation to adapt to limited on-chip caching to avoid significant data transmission losses. They may fall back onto the host CPU to run unsupported layers in the hardware, and may rely on model compilers that support reduced frame sets (such as TFLite)

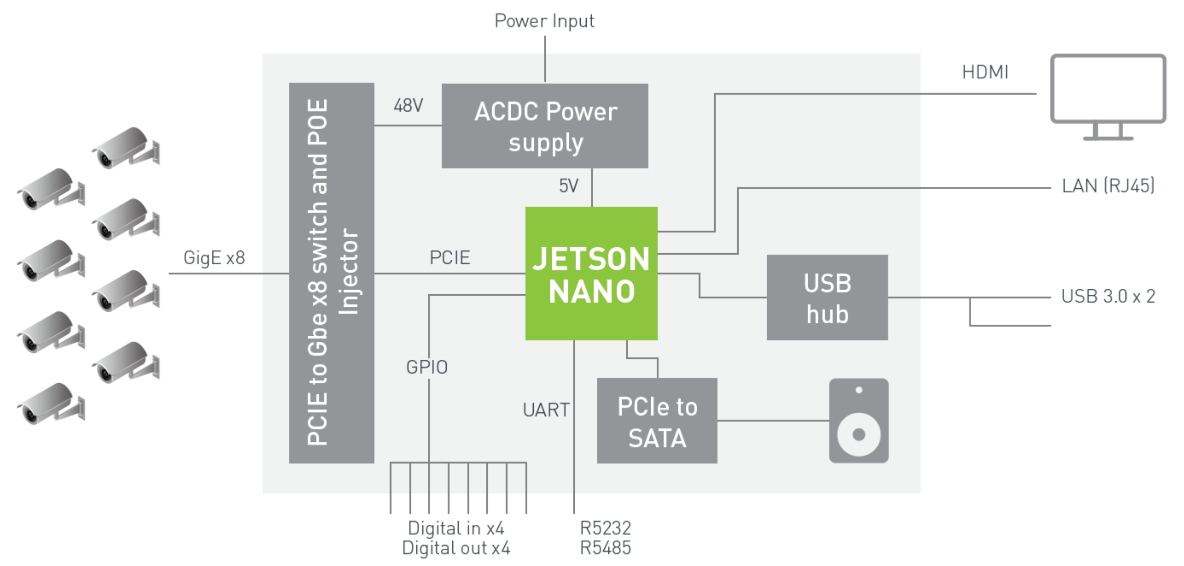
Jetson Nano's flexible software and complete framework support, memory capacity, and unified memory subsystem enable it to run multiple different networks, achieving full HD resolution, including variable batch sizes on multiple sensor streams simultaneously. These benchmark tests represent some examples of popular networks, but users can deploy various models and custom architectures for Jetson Nano by accelerating performance. And Jetson Nano is not limited to DNN inference. Its CUDA architecture can be used for computer vision and digital signal processing (DSP), using algorithms including FFT, BLAS, and LAPACK operations, as well as user-defined CUDA kernels.

3.Multi stream video analysis

Jetson Nano can process up to 8 high-definition fully dynamic video streams in real-time and can be deployed as a low-power edge intelligent video analysis platform for network video recorders (NVRs), smart cameras, and IoT gateways. NVIDIA's DeepStream SDK uses ZeroCopy and TensorRT to optimize end-to-end inference pipelines for optimal performance on edge and local servers. Jetson Nano performs object detection on 8 1080p30 streams simultaneously, and the ResNet based model runs at full resolution with a throughput of 5 million pixels per second (MP/s).

The DeepStream application running on Jetson Nano runs ResNet based object detectors on eight independent 1080p30 video streams simultaneously.

The following block diagram shows an example NVR architecture using Jetson Nano for deep learning analysis to ingest and process up to eight digital streams on a Gigabit Ethernet. The system can decode H.264/H.265 at 500 MP/s and encode H.264/H.265 videos at 250 MP/s.



4.AI Model Training

Developers who want to try training their own models can follow the complete "two day demonstration" tutorial, which covers image classification, object detection, and retraining and customization of semantic segmentation models with transfer learning. Transfer learning can accurately adjust the model weights of a specific dataset and avoid the need to train the model from scratch. Transmission learning is most effectively performed on PC or cloud instances connected to NVIDIA independent GPUs, as training requires more computing resources and time than inference.

However, since Jetson Nano can run complete training frameworks such as TensorFlow, PyTorch, and Caffe, it can also transfer students for those who may not be able to access another dedicated training machine and are willing to wait longer to obtain results. The following figure highlights the preliminary results of PyTorch's two-day transfer learning from training Alexnet and ResNet-18 on a 22.5GB ImageNet subset using Jetson Nano on 200 day images to demonstrate the tutorial:

网络	批量大小	每个时代的时间	张/秒
AlexNet	64	1.16小时	45
RESNET-18	64	3.22小时	16

The time of each period is the time required to fully pass through the training dataset of 200K images. For available results, the classification network may require 2-5 periods, and production models should be trained on discrete GPU systems to obtain more periods until they reach maximum accuracy. However, Jetson Nano allows you to experience deep learning and artificial intelligence on low-cost platforms by retraining the network overnight. Not all custom datasets may be as large as the 22.5GB example used here. Therefore, the image per second represents the training performance of Jetson Nano, the time scaling of each period and the size of the dataset, the training batch size, and network complexity. Other models can also be retrained on Jetson Nano while increasing training time.