



Table of Contents

- 1 Executive Summary
- 2 AI Processing for Camera Image Data
- 3 NVIDIA NGC on Supermicro Validated NGC Ready Systems
- 4 AI/ML Deployment in TensorRT
- 5 Converting ML Model to TensorRT
- 6 Inference Benchmark Results
- 7 Sizing for AI Inference
- 8 Supermicro GPU Servers Specifications
- 9 How to Run NGC
- 10 Guidelines for Model Development
- 11 Additional Training Results
- 12 Support & Services
- 13 Conclusion
- 14 References

White Paper

Supermicro® Systems Powered by NVIDIA GPUs for Best AI Inference Performance Using NVIDIA TensorRT

NVIDIA AI Software from the NGC Catalog for Training and Inference

Executive Summary

Deep learning inferencing to process camera image data is becoming mainstream. With the availability of high-resolution network cameras, accurate deep learning image processing software, and robust, cost-effective GPU systems, businesses and governments are increasingly adopting these technologies. The use cases include retail inventory tracking, on-premise security, insurance claim damage assessment, medical image diagnosis, and many other applications.

This document demonstrates the benefits of using NVIDIA NGC and NVIDIA TensorRT to get the best inference performance using Supermicro systems powered by NVIDIA GPUs. It also shows how to set up NGC on a Supermicro server and how to use TensorRT for inference. The primary focus of this paper is about the key capabilities of Supermicro systems powered by NVIDIA GPUs for inference. Benchmark data that Supermicro engineers collected, sizing recommendations, and server selection for inference deployment are also included in this paper.

KEY CUSTOMER BENEFITS

- Optimized IT deployment ROI
- Faster AI results
- Effective scaling
- Faster deployment
- Cost-effective
- Energy-savings

AI Processing of Camera Image Data

In the past ten years, there have been significant advances in the recognition and classification of camera image data with neural-network-based AI. It began with a relatively simple Convolution Neural Network (CNN), followed by more advanced AI models. Recognition accuracy has advanced beyond what a human can do. By combining classification, segmentation, labeling, and other image feature extraction techniques, these AI models are now being applied to business applications, including the following:

- Retail inventory tracking
- Retail self-checkout
- Self-driving cars
- On-premise security and monitoring
- Automatic insurance claim damage assessment
- Medical diagnosis
- And many other applications

The general approach is to train an AI model with labeled data using a single NVIDIA GPU system or a cluster of GPU systems. The AI training could take hours to days, depending on the amount of data, the number of times (epochs) data needs to be examined, and types of systems used. Once an AI model is trained to the required accuracy, then it is deployed in one or more applications to make the AI inference of new incoming data.

Supermicro Validated NGC-Ready Systems

[NVIDIA NGC](#) makes the entire AI development and deployment process much more methodical and manageable. The NGC Catalog provides regularly updated AI software containers, pre-trained models, helm charts, use-case specific collections and industry-focused application frameworks. The pre-trained models could be used to do inferencing immediately, or they may be re-trained via transfer learning with custom data to tune the model to specific needs. Useful tools are available in NGC, such as transfer learning toolkits, industry-specific frameworks, the NVIDIA DeepStream SDK to process video data, and the AI inference tool, NVIDIA TensorRT.

The NGC software stack can be deployed on multiple Supermicro validated NGC-Ready Systems. NVIDIA NGC Support Service is available to help the customer get started with NGC, address questions, and resolve problems in the use of NGC.

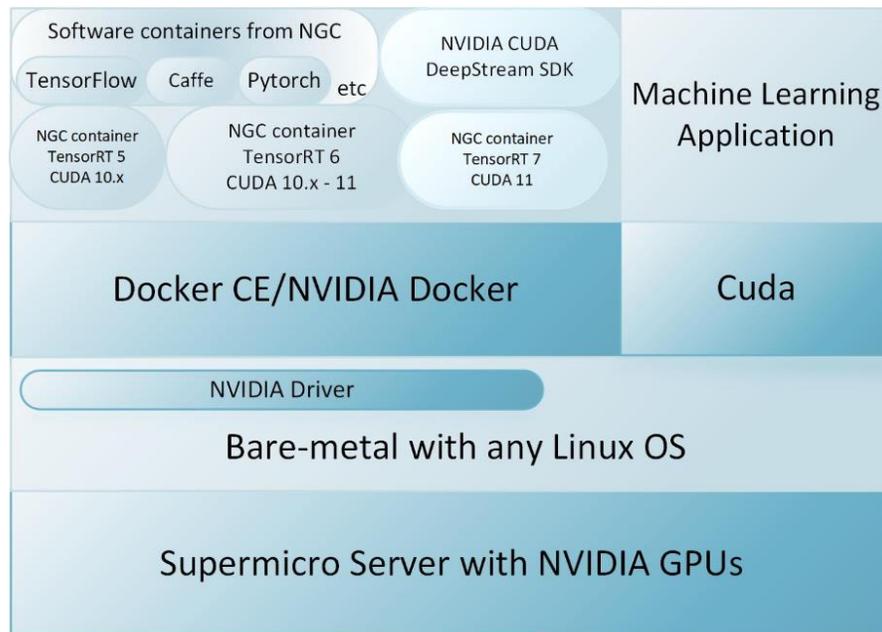


Figure 1. NVIDIA Software Stack on a Supermicro System

The software stack includes the following:

- NVIDIA Driver is the software driver for NVIDIA Graphics GPU installed in the Linux Operating System (Ubuntu, RHEL, or CentOS) running on Supermicro systems.
- NVIDIA CUDA is a parallel computing platform and application programming interface (API) model that works in conjunction with the NVIDIA Driver. With CUDA, developers can dramatically speed up computing applications by taking full advantage of the GPU's parallel processing power.
- NVIDIA Docker to manage NGC containers. To run the NGC containerized applications, install both Docker CE and NVIDIA-Docker. Docker CE is a free and open-source containerization platform. NVIDIA-Docker is required to run the NVIDIA related containers. NVIDIA Docker is a wrapper around the Docker CLI that transparently provisions a container with the necessary dependencies to execute code on the GPU.
- Containerized deep learning, machine learning, and HPC applications from the NGC Catalog.
- Supermicro Validated NGC-Ready systems enable customers to purchase an NGC Support Services contract and gain access to their own private registry to share, collaborate, and deploy software.

AL/ML Deployment in NVIDIA TensorRT

TensorRT for Inferencing

TensorRT is an NVIDIA specific Inferencing Engine, which provides APIs and parsers to import trained models from all major deep learning frameworks like TensorFlow, Caffe, PyTorch, ONNX, Matlab, Mxnet, and a few others, and convert them to a TensorRT engine and run inference algorithms. An NGC container comes with predefined CUDA core, TensorRT, all other necessary AL/ML frameworks, and their dependencies.

The following diagram is a simple explanation of how machine learning code developed from TensorFlow is converted to a TensorRT engine for faster Inferencing, to be able to predict a given image. Take any trained TensorFlow model and convert it to a frozen graph; this step allows freezing all the graphs and weights in a single file. Follow NVIDIA's instructions to install the TensorRT; at this point, use TensorRT API to convert the above TensorFlow frozen graph created to a Tensorflow TensorRT (TF-TRT) optimized model for further prediction.

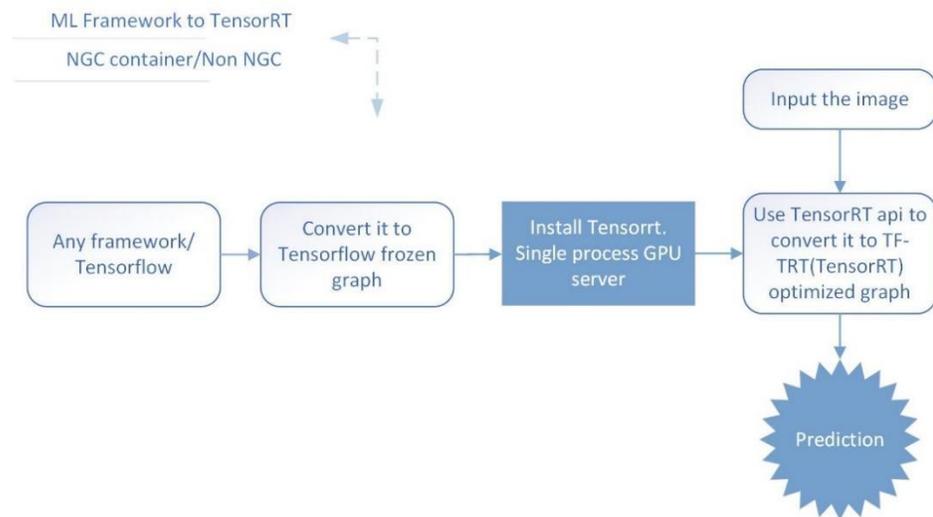


Figure 2. Machine learning model deployment in TensorRT

Converting ML Model to TensorRT Engine

TensorRT is used for Inferencing. It is necessary to convert the existing ML models to TensorRT format to leverage NVIDIA GPUs for high performance. There are various ways to transform an ML model to TensorRT; a few references are listed below, so you can go over how to get the best throughput from TensorRT.

Using our internal demo, the trained ML Model from Keras/TensorFlow is converted to run on the TensorRT engine. With TensorRT 5.1.5, it took about 1 to 2 seconds overall to make a single prediction, whereas the non-TensorRT took about 8 to 9 seconds to do the same Inferencing.

The widely used model is TensorFlow with TensorRT (TF-TRT) optimization, listed in the reference section. Similarly, an ONNX model can be converted to run on TensorRT for getting optimized results.

PyCUDA API is just another way to convert any ML framework like TensorFlow to TensorRT, which enables data scientists to access NVIDIA's CUDA parallel computation using Python and C++. The URL provided in the reference section specifies instructions to convert a TensorFlow model to an UFF (unified) model, and further to a TensorRT engine. A data scientist can now transfer the input data, using the TensorRT engine context to the GPU device from the host machine, and get a prediction.

Inference Benchmark Results

Running the inferencing benchmark inside an NGC container with a framework such as TensorFlow, Caffe, and TensorRT6, the throughput on a NVIDIA V100 is significantly better than an NVIDIA T4 GPU. Googlenet processed 11,111 images/sec, and Resnet50 processed 7142 images/sec, respectively. Refer to figure 3 for testing results.

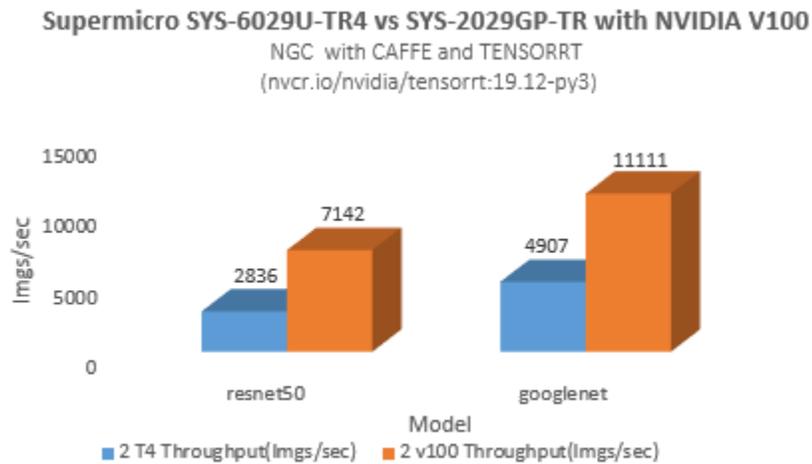


Figure 3. Inference using NGC to determine the benchmarking (performance of single T4 with INT8, single V100 with FP16)

Sizing for AI Inference

AI inference to process image data can be performed in the data center core or the edge. The amount of data and the level of the required precision and the speed of AI inference determine the type of systems needed for either location. The benchmark results help us size the most appropriate systems for optimal deployment.

It is recommended to run sizing tests before determining the exact number of systems to deploy. For other image models, please check with Supermicro.

Supermicro Servers Powered by NVIDIA GPUs

The following three figures are the sample system configurations used for NGC training and Inferencing benchmarks.

Items	Description	Qty
SYS-6049GP-TRT	GPU 4U 1-NodeDP 24-DIMM 8*2.5 HS SATA 2-10GbE	1
P4X-SKL6150-SR37K	SKL-SP 6150 18C/36T 2.7G 24.75M 10.4GT UPI	2
MEM-DR412LCL02-LR26	128GB DDR4 LRDIMM 2666 ECC REG	4
HDD-A16TST16000NM002G	3.5", 16TB, 7.2K RPM, SAS3 12Gb/s, Cache 256MB, 512e/4Kn	2
HDS-S2T1-MZ7LH7T6HMLA05	PM883, 7.6T, SATA 6Gb/s, V4 TLC VNAND, 2.5", 7mm	5
GPU-NVTT4	NVIDIA Tesla T4 16GB GDDR6 PCIe 3.0—Passive Cooling 70	2

Figure 4. SYS-6029U-TR4 with 2 NVIDIA T4 GPUs

Items	Description	Qty
SYS-2029GP-TR	X11DPG-SN x1	1
P4X-SKL6150-SR37K	SKL-SP 6150 18C/36T 2.7G 24.75M 10.4GT UPI	2
MEM-DR432LSL03-ER26	32GB DDR4 1.2V 2666 ECC REG	6
HDS-X2A-XS7680TE70004	2.5" Lange 7.68TB SAS 12Gb/s, 15mm, 2.5", 0.8DWPD SSD.HF.RoHS	2
HDD-2A2000-ST2000NX0353	2.5", 2TB, SAS3.0 12GB/S, 7.2K RPM, 512E 128M, SEDFIPS	5
AOC-STG-I4T-O	4 Port 10Gbps NIC	1
GPU-NVTV100-32	NVIDIA TeslaV100 32GB CoWoS HBM2 PCIe 3.0—PassiveCoolin	2

Figure 5. SYS-2029GP-TR with 2 NVIDIA V100 GPUs

Items	Description	Qty
SYS-4029GP-TVRT	X11DGO-T x1	1
P4X-SKL8180-SR377	SKL-SP 818028C/56T 2.5G 38.5M 10.4GT UPI	2
MEM-DR412LCL02-LR26	128GB DDR4LRDIMM 2666 ECCREG	4
HDD-2A2400-AL15SEB24EQ	2.5" 2.4TB SAS312Gb/s 10K RPM 128MB 512e	2
HDS-T2AKPM51RUG7T68	PM5 7.68TB SAS 12Gb/s 15mm BiCS3 eTLC 1DWPD	5
AOC-S25GB2S-O	[NR]AOC-S25GB2S (Retail Pack)	1
GPU-NVTV100-32-SXM2	NVIDIA® Tesla®V100 SXM2 32GB CoWoS HBM2. NVLink	8

Figure 6. SYS-4029GP-TVRT With 8 NVIDIA V100 GPUs

How to Run NGC

There are several ways to run NGC:

- A.** Get a Supermicro validated NGC Ready System, with the NGC software platform and operating system pre-installed from the factory, or done by a Supermicro partner/ reseller.
- B.** Run NGC in vComputeServer on top of a supported Hypervisor such as VMware vSphere.
- C.** Install on bare-metal Supermicro server running a supported operating system, Ubuntu 18.04 or Red Hat Enterprise Linux 7 or CentOS 7 (no enterprise support for CentOS). Use the following steps:
 - Install the NVIDIA driver.
 - Setup Docker runtime engine, and NVIDIA Docker.
 - Docker: pull the AI optimized deep learning NGC Image, for example, "nvcr.io/nvidia/tensorflow:19.12-tf1-py3".
 - Create the Docker container from the above image, and execute the model for benchmarking.

Here is the sample command to train a resnet model in a NGC container:

```
nvidia-docker run -it --rm -v $(pwd):/work -w /workspace/nvidia-examples/cnn nvcr.io/nvidia/tensorflow:19.12-tf1-py3 mpiexec --allow-run-as-root -np 1 --bind-to socket python -u ./resnet.py --batch_size 256 --num_iter 1000 --precision fp16 --iter_unit batch --layers 50
```

The following figure shows the general procedure to run the NGC benchmark.



Figure 7. How to run the NGC

Guidelines for AI Development

By using NGC, there are several quick ways to get AI workload started.

- A. Pre-trained Models
- B. Transfer Learning
- C. Optimizing Models
- D. It is best to follow NVIDIA guidelines if data scientists decide to develop their models and later use the TensorRT for Inferencing. NVIDIA provides a list of supported layers that follow guidelines for model development, and the link provided in the reference section. This website is updated from NVIDIA, lists all the features/precisions supported by different layers of neural network in TensorRT, and supported CUDA versions.

Additional Benchmark Results on AI Training

While this document focuses on AI inference, there are impressive AI training results on Supermicro systems running with 1, 2, and 8 NVIDIA V100 GPUs. The scaling from 1 to 2 in a single system scale linearly. There is also linear scaling in performance comparing the result from the 2-GPU system (SYS-2029GP-TR) to the result from the 8-GPU system (SYS-4029GP-TVRT).

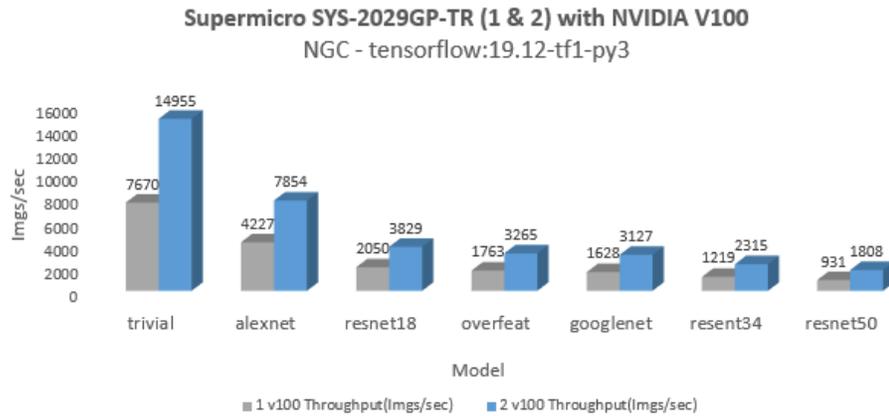


Figure 8. Training using TensorFlow container from the NGC Catalog to determine the benchmarking

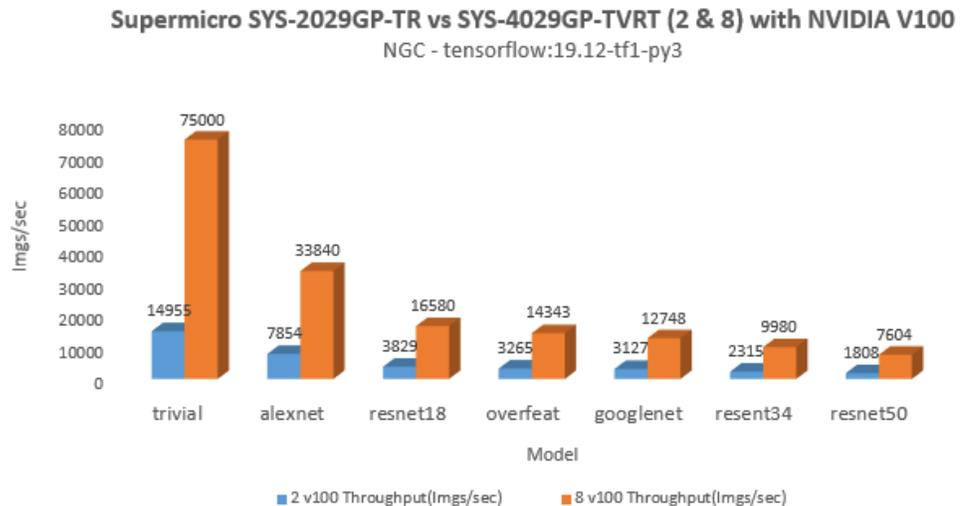


Figure 9. Training using TensorFlow container from the NGC Catalog to determine the benchmarking

Support and Services

Supermicro offers the following support services to help customers run NGC on the Supermicro validated NGC Ready systems.

1. Hardware system support.
2. Linux OS support, available as part of the enterprise subscription for the Ubuntu or the Red Hat Enterprise Linux Operating System
3. NVIDIA NGC Support Services.

Conclusion

This paper focused on AI inference and deployment, coupled with some training results. Inference performance is significantly higher by using TensorRT. Supermicro AI benchmarks provide excellent guidance on sizing systems for AI deployment. Customers can choose to use low power NVIDIA T4 or high-performance NVIDIA V100. These GPUs are available on the Supermicro validated NGC Ready Systems.

Please contact Supermicro sales representative for more information.

For More Information

- <https://www.supermicro.com/en/products/system/4U/4029/SYS-4029GP-TVRT.cfm>
- <https://www.supermicro.com/en/products/GPU>
- <https://www.supermicro.com/en/solutions/ai-deep-learning>



REFERENCES

NVIDIA TensorRT team has a list of supported layers, that lets you follow guidelines for model development.

<https://docs.NVIDIA.com/deeplearning/TensorRT/support-matrix/index.html>

There are a couple of methods to convert Tensorflow to TRT

<https://docs.NVIDIA.com/deeplearning/frameworks/tf-trt-user-guide/index.html#using-savedmodel>

<https://docs.NVIDIA.com/deeplearning/frameworks/tf-trt-user-guide/index.html#TensorRT-plan>

Any model can be converted to the ONNX model and imported into TensorRT for Inferencing.

<https://devblogs.NVIDIA.com/speeding-up-deep-learning-inference-using-TensorRT/>

About Super Micro Computer, Inc.

Supermicro® (NASDAQ: SMCI), the leading innovator in high-performance, high-efficiency server technology is a premier provider of advanced Server Building Block Solutions® for Data Center, Cloud Computing, Enterprise IT, Hadoop/Big Data, HPC and Embedded Systems worldwide. Supermicro is committed to protecting the environment through its "We Keep IT Green®" initiative and provides customers with the most energy-efficient, environmentally-friendly solutions available on the market.

Learn more at www.supermicro.com

No part of this document covered by copyright may be reproduced in any form or by any means — graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system — without prior written permission of the copyright owner.

Supermicro, the Supermicro logo, Building Block Solutions, We Keep IT Green, SuperServer, Twin, BigTwin, TwinPro, TwinPro², SuperDoctor are trademarks and/or registered trademarks of Super Micro Computer, Inc.

Intel, the Intel logo, and Xeon are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. All other brands names and trademarks are the property of their respective owners.

© Copyright Super Micro Computer, Inc. All rights reserved.

Printed in USA

 Please Recycle

000_WP-Template_Rev6

