Technical Report

NetApp ONTAP AI Reference Architecture for Healthcare: Diagnostic Imaging

Solution Design

Rick Huang, Sung-Han Lin, Sathish Thyagarajan, NetApp
Jacci Cenci, NVIDIA

March 2020 | TR-4811

In partnership with

Abstract

This reference architecture offers guidelines for customers building artificial intelligence (AI) infrastructure using NVIDIA DGX-2™ systems and NetApp® AFF storage for healthcare use cases. It includes information about the high-level workflows used in the development of deep learning (DL) models for medical diagnostic imaging, validated test cases, and results. It also includes sizing recommendations for customer deployments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Executive Summary

The NVIDIA DGX™ family of systems is made up of the world's first integrated AI systems that are purpose-built for enterprise AI. NetApp® AFF storage systems deliver extreme performance and industry-leading hybrid cloud data-management capabilities. NetApp and NVIDIA have partnered to create the NetApp ONTAP® AI reference architecture, a turnkey solution for AI and machine learning (ML) workloads that provides enterprise-class performance, reliability, and support.

This technical report gives directional guidance to healthcare providers building AI infrastructure in support of diagnostic imaging practices in hospitals. It includes information about the high-level workflows used in the development of DL data pipeline models for medical imaging, validation test cases and results, and sizing recommendations for deployments. The tests were performed using an NVIDIA DGX-2 system and a NetApp AFF A800 storage system.

The target audience for the solution includes the following groups:

- Enterprise architects who design solutions for the development of AI models and software for healthcare use cases such as medical image segmentation
- Data scientists looking for efficient ways to achieve DL development goals
- Data engineers in charge of maintaining and processing healthcare data such as medical imaging and electronic health records
- Executive and IT decision makers and business leaders interested in transforming the healthcare experience and achieving the fastest time to market from AI initiatives

2 Solution Overview

2.1 Medical Imaging Use Case Summary

Advances in medical imaging technologies, including 3D and 4D capabilities, real-time analytics, and GPU-accelerated processing give radiologists powerful tools to make faster and more accurate diagnoses and recommendations for care. Specifically, semantic segmentation algorithms are becoming increasingly general purpose and translatable to new tasks and use cases. A model that works out of the box on various diagnostic and medical imaging tasks would have a tremendous impact in healthcare.

Healthcare workloads can contain a wide variety of data types, including the following:

- Electronic health records
- Surgical robot videos
- Different imaging modalities
  - Retinal images
  - Ultrasound imaging
  - Computed tomography (CT)
  - Positron emission tomography (PET)
  - Magnetic resonance imaging (MRI)

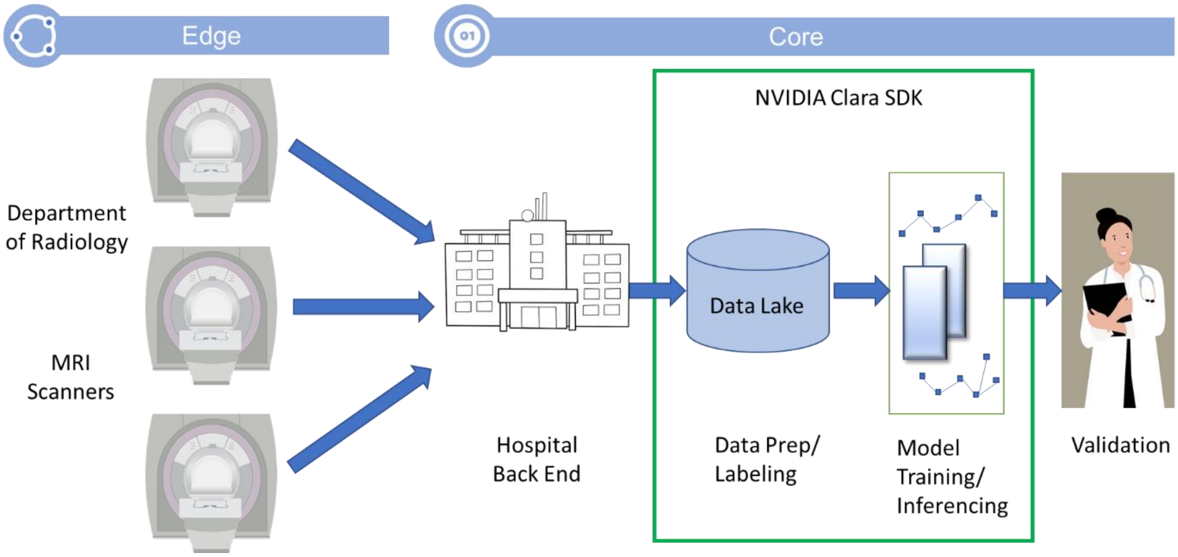
All this data contributes to different aspects of the healthcare services like medical imaging, digital pathology, genomics, and many others. Model training requirements vary for different data types, and achievable performance on compute and storage resources can also vary. The goal is always to saturate the GPUs and provide the highest throughput at the lowest latency from the storage side. This technical report addresses the challenges in the training phase, which involves delivering best-in-class performance to reduce the time to insight and increase accuracy. It also discusses validation of the AI and DL model training for hippocampus segmentation using a publicly available dataset with the NVIDIA Clara™ platform. See Section 4 for the testing details.

The hippocampus is a vital component of the human brain. It plays an important role in the consolidation of information from short-term memory to long-term memory and in spatial memory that enables navigation. In Alzheimer's disease and other forms of dementia, the hippocampus is one of the first regions of the brain to suffer damage. Accurate identification of the hippocampus in an MRI is an important step in the process of diagnosis. However, it can be difficult for radiologists and other doctors to segment two small, neighboring structures with high precision. DL models can perform this work much faster and more accurately, allowing medical practitioners to spend more time on patient diagnosis and care and less time examining images.

NVIDIA Clara is a computational platform that enables developers to build, manage, and deploy intelligent medical imaging workflows. The NVIDIA Clara Train SDK™ offers state-of-the-art tools and technologies that accelerate data annotation, adaptation, and development of AI models for healthcare imaging workflows. This validation uses the NVIDIA Clara platform to provide AI-assisted annotation to label a publicly available brain imaging dataset and train a hippocampus segmentation model based on the ResNet-50 and AH-Net architectures.

Figure 1 illustrates a high-level overview of the medical imaging solution. Medical images are collected at edge sites such as MRI scanners in radiology departments. Images are then collated in a data lake at a core facility, such as a hospital's back-end servers. Next, the NVIDIA Clara SDK enables efficient data preparation, labeling, model training, and inferencing. Finally, radiologists or other doctors validate the output.

Figure 1) Proposed medical imaging use case solution overview.



3 Solution Technology

The NetApp ONTAP AI architecture, powered by NVIDIA DGX systems and NetApp cloud-connected storage systems, was developed and verified by NetApp and NVIDIA. This reference architecture gives IT organizations the following advantages:

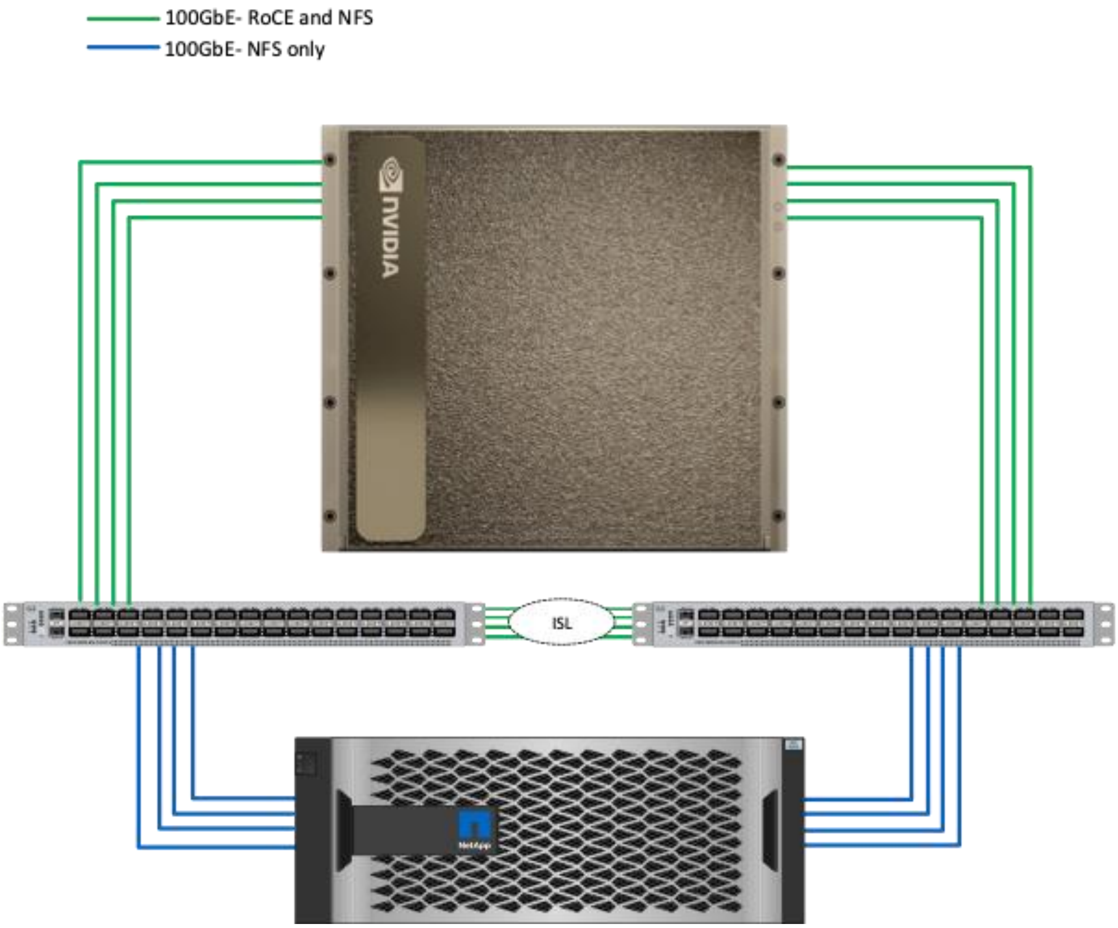
- Eliminates design complexities
- Allows independent scaling of compute and storage
- Enables customers to start small and scale seamlessly
- Offers a range of storage options for various performance and cost points

NetApp ONTAP AI tightly integrates DGX systems and NetApp AFF A800 storage systems with state-of-the-art networking. NetApp ONTAP AI and DGX systems simplify artificial intelligence deployments by eliminating design complexity and guesswork. Customers can start small and grow their systems in an uninterrupted manner while intelligently managing data from the edge to the core to the cloud and back.

The AFF A800 storage system has been verified with nine DGX-1 systems and three DGX-2 systems. Furthermore, by adding more network switches and storage controller pairs to the ONTAP cluster, the architecture can scale to multiple racks to deliver extremely high throughput, accelerating training, and inferencing. With this flexible approach, the ratio of compute to storage can be altered independently based on the size of the data lake, the models that are used, and the required performance metrics. For detailed information about ONTAP AI with DGX-1 systems, see the NetApp Verified Architectures NVA-1121 and NVA-1138. For information about ONTAP AI with DGX-2 systems, see NVA-1135.

This solution was validated with one NetApp AFF A800 system, one NVIDIA DGX-2 system, and two Cisco Nexus 3232C 100 Gigabit Ethernet (100GbE) switches. As illustrated in Figure 2, the DGX-2 system is connected to the Nexus switches with eight 100GbE connections that are used for inter-GPU communications by using remote direct memory access (RDMA) over Converged Ethernet (RoCE). This configuration allows additional DGX-2 systems to be added as needed. Traditional IP communications for NFS storage access also occur on these links. Each storage controller is connected to the network switches with four 100GbE links.

Figure 2) ONTAP AI healthcare solution topology.



3.1 Hardware Requirements

This solution was validated using one DGX-2 system and one AFF A800 storage system. The test results shown in Section 4.6 are based on this configuration and do not represent the maximum performance that the full-scale architecture can achieve.

Table 1 lists the hardware components that are required to implement the solution as tested. The hardware components used in a specific customer implementation should be based on the sizing guidance in Section 5.

Table 1) Hardware requirements.

Hardware	Quantity
NVIDIA DGX-2 system	1
NetApp AFF A800 storage system	1 high-availability (HA) pair; includes 2 controllers and 48 NVMe SSDs (3.8TB or larger)
Cisco Nexus 3232C network switch	2

3.2 Software Requirements

Table 2 lists the software components that are required to implement the solution as tested.

Table 2) Software requirements.

Software	Version or Other Information
NetApp ONTAP data management software	9.6
Cisco NX-OS switch firmware	7.0(3)I6(1)
NVIDIA DGX OS	4.0.4 - Ubuntu 18.04 LTS
Clara Train SDK	v2.0
Medical Imaging Interaction Toolkit (MITK) Workbench	2018.04.2 with NVIDIA AI-Assisted Annotation v1.0.1
Docker container platform	18.06.1-ce [e68fc7a]
Container version	nvcr.io/nvidia/clara-train-sdk:v2.0
Machine learning framework	TensorFlow 1.13.3
Open MPI	3.1.3

4 Hippocampus Segmentation Data Annotation and Model Training

The model training performance of this solution was validated using the Information eXtraction from Images (IXI) brain imaging dataset with AI-assisted annotation by the NVIDIA Clara Train SDK. The following sections contain more information about this dataset and the testing results.

4.1 IXI Brain Imaging Dataset

The IXI project collected nearly 600 MR images from normal, healthy subjects. The MR image acquisition protocol for each subject includes the following:

- T1, T2, and PD-weighted images

- Magnetic resonance angiography (MRA) images
- Diffusion-weighted images in 15 directions

The images are in NIFTI format with a total size of 56GB. The NVIDIA Clara Train SDK for AI-assisted annotation was used to create labeled data for the hippocampus segmentation model training.

4.2 NVIDIA Clara Train SDK

The NVIDIA Clara Train SDK runs on top of TensorFlow and provides AI-assisted annotation to multiple medical imaging viewers. The SDK contains pretrained AI models and development tools to accelerate creation of AI algorithms for medical imaging workflows. The NVIDIA Clara Train SDK consists of the following:

- AI-assisted annotation APIs for auto annotation and interactive annotation
- The Annotation Server, which provides pretrained models to the client application for transfer learning
- Unified, Python-based APIs that enable techniques like transfer learning for training models and the ability to train from scratch
- Pretrained models based on AH-Net, DenseNet, ResNet-50, and Dextr3D, which are packaged as complete 2D or 3D model applications for segmentation, classification, and annotation

The following pretrained models (downloaded from the NVIDIA Model Registry) are available in the NVIDIA Clara Train SDK for specific classification and segmentation use cases:

- Brain tumor segmentation
- Liver and tumor segmentation
- Lung tumor segmentation
- Spleen segmentation
- Chest X-ray classification

In this technical report, we use a hippocampus segmentation model because the IXI brain imaging dataset contains only healthy subjects. There are no brain tumor images available in the IXI dataset MR images. Therefore, it would be counterproductive to employ the brain tumor segmentation model in this situation. The next section provides details on how to develop your own model in NVIDIA Clara.

4.3 Bring Your Own Components in NVIDIA Clara Train SDK

The NVIDIA Clara Train SDK allows users to solve new problems and innovate by writing their own components in a modular way. To do this, users can write their own components in Python files, and then point to these files in the `train_config.json` file by providing the paths for the new components.

Users can add the following list of different components:

- Data pipelines
- Models
- Loss functions
- Optimizers
- Metrics

Users can use the predefined models offered by NVIDIA or choose their own model architecture when configuring a training workflow, provided that the model adheres to the following model API specification:

```
import tensorflow as tf
from ai4med.common.graph_component import GraphComponent
from ai4med.common.build_ctx import BuildContext

class Model(GraphComponent):
    """Base class of Models

    Args:
        None
    Returns:
        Prediction results
    """

    def __init__(self):
        GraphComponent.__init__(self)

    def get_loss(self):
        """Get the additional loss function in AHNet model.

        Args:
            None

        Returns:
            Loss function
        """
        return 0

    def get_update_ops(self):
        """Get the update_ops for Batch Normalization.

        The method "tf.control_dependencies" allow the operations used as inputs
        of the context manager are run before the operations defined inside the
        context manager. So we use "update_ops" to implement Batch Normalization.

        Args:
            None

        Returns:
            Update operations
        """
        return tf.get_collection(tf.GraphKeys.UPDATE_OPS)

    def get_predictions(self, inputs, is_training, build_ctx: BuildContext):
        """Forward computation process of model for both training and inference.

        Args:
            inputs (tf.Tensor): input data for the AHNet model
            is_training (bool): in training process or not
            build_ctx(BuildContext): reserved argument for future features

        Returns:
            Prediction results
        """
        raise NotImplementedError('Class {} does not implement get_predictions'.format(
            self.__class__.__name__))


    def build(self, build_ctx: BuildContext):
        """Connect model with graph.

        Args:
            build_ctx: specified graph context

        Returns:
            Prediction results
        """

        inputs = build_ctx.must_get(BuildContext.KEY_MODEL_INPUT)
        is_training = build_ctx.must_get(BuildContext.KEY_IS_TRAIN)
        return self.get_predictions(inputs, is_training, build_ctx)
```

By extending the `Model` class, we can implement a hippocampus segmentation model to process the IXI brain imaging dataset.

4.4 Data Curation and AI-Assisted Annotation

Because the IXI brain imaging dataset contains different MR image acquisition protocols, you must curate the data to adhere to the same protocol so that the variability between different protocols does not confuse the model. In other words, it's better to have different models for T1, T2, PD-weighted, MRA, and diffusion-weighted images rather than training a model to handle the different protocols simultaneously. For this reason, we used T1 images for AI-assisted annotation and transfer learning. Note that the procedures and tests described in this report work on other protocols as well.

AI-assisted annotation uses DL techniques to take points of interest identified by radiologists along with the 3D volume data as input to create an auto-annotated set of slices. After loading either a pretrained DL model from NVIDIA or models developed by users, in our case the hippocampus segmentation, you can perform AI-assisted annotation one input image at a time. Issue the following command to start the NVIDIA Clara AI-Assisted Annotation (AIAA) Server:

```
root@dgx2-1:~# docker run --runtime=nvidia \
>    --mount type=bind,source=/mnt/mount_0/dataset/,target=/workspace/data\
>    -it --rm -p 5000:5000 \
>    nvcr.io/nvidia/clara-train-sdk:v2.0 \
>    start_aas.sh
```

Using the NVIDIA Clara AIAA Server, the hippocampus segmentation model can be loaded by issuing a `curl` command.

First you must compress the TensorFlow model before loading it into the AIAA server. The `zip` and `curl` commands are as follow:

```
# Zip the checkpoint files
root@dgx2-1:~#zip model.zip \
>    model.ckpt.data-00000-of-00001 \
>    model.ckpt.index \
>    model.ckpt.meta

root@dgx2-1:~#curl -X PUT "http://0.0.0.0:5000/admin/model/annotation_mri_hippocampus" \
>     -F "config=@config_aiaa.json;type=application/json" \
>     -F "data=@model.zip"
```

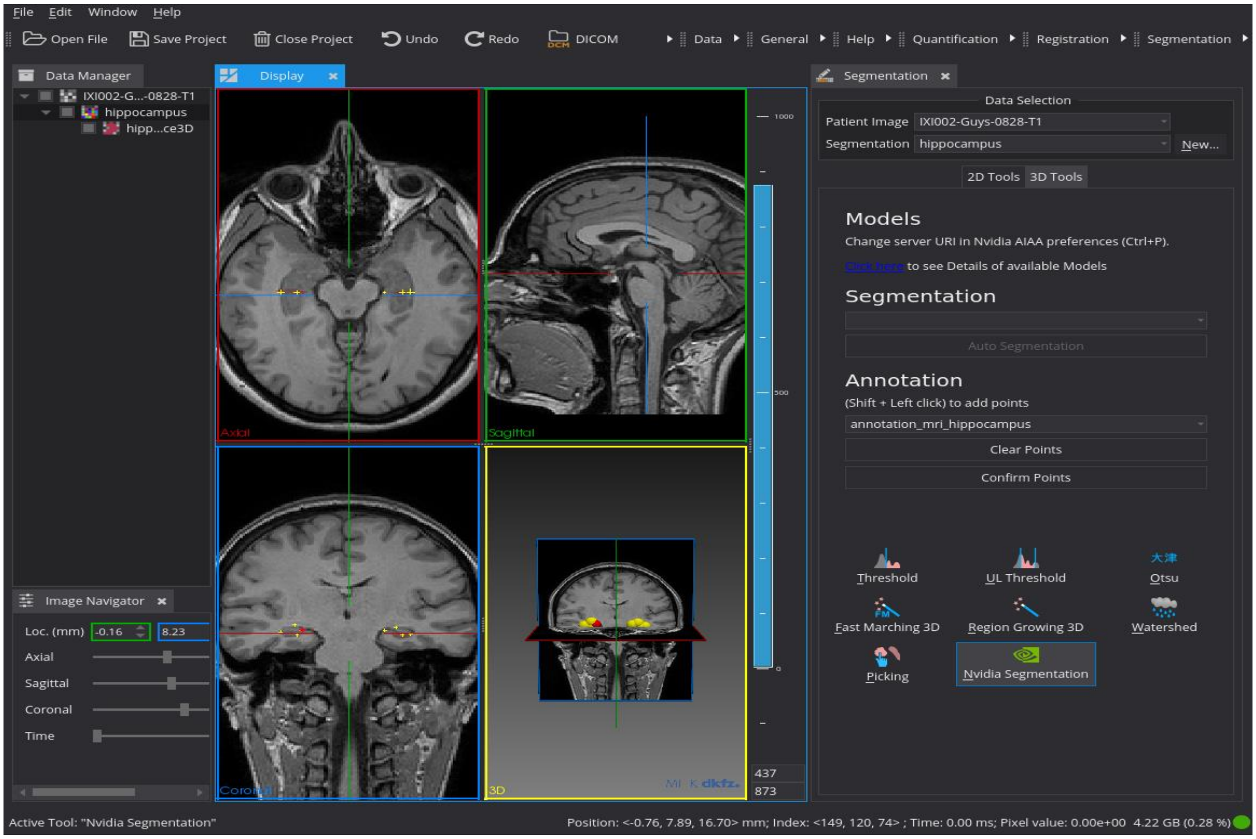
Data labeling for individual MR images is performed in the MITK, a free, open-source software system for developing interactive medical image processing. 3D Slicer client integration is also available with the NVIDIA Clara Train SDK. After you install the required libraries, such as `QT5`, `libgl`, `libxt`, `libtiff`, and `libxkbcommon`, the prebuilt binary is executed to start MITK Workbench.

Note: Most users install MITK locally and then point to the container running on a remote server. If you choose to connect to a remote server with SSH and then start MITK there, the following setting is crucial to avoid an error when you start MITK Workbench:

```
export QTWEBENGINE_DISABLE_SANDBOX=1
```

When started successfully, MITK Workbench with the NVIDIA Segmentation Tool resembles Figure 3.

Figure 3) MITK workbench with NVIDIA segmentation tool.



Patient Image IXI002-Guys-0828-T1

Segmentation hippocampus

Change server URI in Nvidia AIAA preferences (Ctrl+P).

Click here to see Details of available Models

(Shift + Left click) to add points

annotation_mri_hippocampus

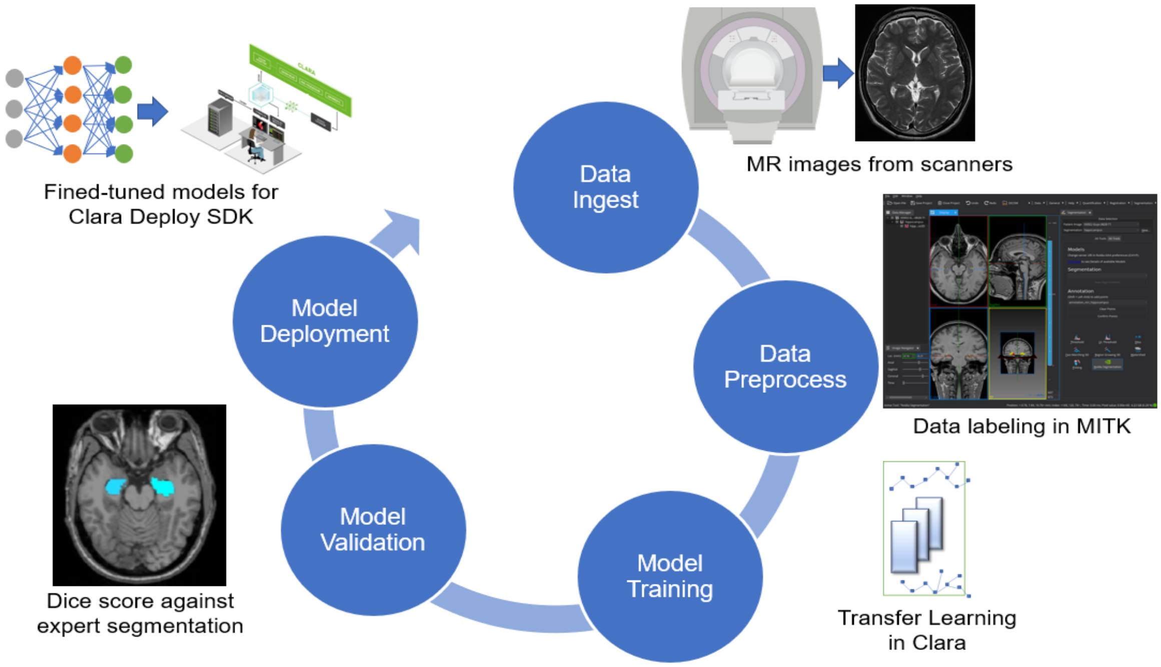
Active Tool: "Nvidia Segmentation"

Position: <-0.76, 7.89, 16.70> mm; Index: <149, 120, 74> ; Time: 0.00 ms; Pixel value: 0.00e+00 4.22 GB (0.28 %)

4.5 Data Pipeline Architecture

Figure 4 illustrates the DL development cycle view of the proposed medical imaging workflow. First, data is ingested from MR scanners. In this report, the data is unlabeled MR images from the IXI dataset. In the preprocessing phase, the hippocampus segmentation model is loaded into the AIAA Server, and MITK Workbench is used to perform AI-assisted annotation. After annotating brain MR images, data is labeled for transfer learning. The NVIDIA Clara Transfer Learning Toolkit is then used to produce a finely tuned model to achieve better segmentation accuracy. Next, the mean Dice score is calculated against human expert segmentation to validate the model. Finally, the model can be deployed at edge sites by using the NVIDIA Clara Deploy SDK.

Figure 4) Deep learning development cycle view of the proposed medical imaging solution.



4.6 Transfer Learning Model Training

The Transfer Learning Toolkit in NVIDIA Clara Train is a Python-based SDK that enables developers to leverage optimized, ready-to-use, pretrained models. These models, available from the NVIDIA GPU CLOUD™ (NGC) Model Registry, accelerate the developer's DL training process, reducing the costs associated with large-scale data collection, labeling, and training models from scratch.

The toolkit offers an end-to-end workflow for accelerating DL training and inference for medical imaging use cases. The models provided are fully trained for medical-imaging-specific use cases such as organ and tumor segmentation and classification. In this technical report, we used a model for volumetric (3D) segmentation of the hippocampus head and body from mono-modal MRI images. The model was trained using a pipeline that was awarded a runner-up award in the Medical Segmentation Decathlon Challenge 2018.

The hippocampus segmentation model is a 3D anisotropic hybrid network (AH-Net) that learns informative features for object detection and segmentation tasks in 3D medical images. The 3D AH-Net is obtained by first training a 2D, fully convolutional ResNet-50 network, and then extending the 2D kernel with one additional dimension to transform the 2D network feature encoder into a 3D network. A feature decoder subnetwork is added to extract the 3D context. Finally, a pyramid volumetric pooling module is placed at the end of the network before the final output layer. For detailed information about 3D AH-Net, see the paper 3D Anisotropic Hybrid Network. AH-Net is superior for capturing spatial correlations in 3D images, compared to DenseNet or ResNet-50, which were developed for 2D images.

After 260 MR images from the IXI dataset were labeled, the data was used to fine-tune the hippocampus segmentation model. To reflect the hardware demands of a common DL training workload, for example with the ResNet-50 network in which 1.2 million training images were used, the 260 labeled images were duplicated 5,000 times to obtain a training set of 1,300,260 images. This data was then used to train the transfer learning model. The model developed for the Medical Segmentation Decathlon Challenge 2018 was reused as the starting point for a model tailored to the IXI brain imaging dataset. Network throughput, CPU utilization from the NetApp AFF A800, and CPU and GPU usage from the DGX-2 system were recorded during the transfer-learning-model training.

You can start the NVIDIA Clara Train v2.0 container by issuing the following command:

```
docker run --runtime=nvidia -it -v /var:/var -v /mnt:/mnt -v /raid:/raid -v /tmp:/tmp --rm
nvcr.io/nvidia/clara-train-sdk:v2.0 /bin/bash
```

The following script performs Horovod-based training with 16 GPUs, which fully utilizes the computational power of a single DGX-2 system:

```
train_16gpu.sh

#!/usr/bin/env bash

my_dir="$(dirname "$0")"
. $my_dir/set_env.sh

echo "MMAR_ROOT set to $MMAR_ROOT"

# Data list containing all data
CONFIG_FILE=config/config_train.json
ENVIRONMENT_FILE=config/environment.json

mpirun -np 16 -H localhost:16 -bind-to none -map-by slot \
    -x NCCL_DEBUG=INFO -x LD_LIBRARY_PATH -x PATH -mca pml ob1 -mca btl
^openib --allow-run-as-root \
    python3 -u  -m nvmidl.apps.train \
    -m $MMAR_ROOT \
    -c $CONFIG_FILE \
    -e $ENVIRONMENT_FILE \
    --set \
    DATASET_JSON=$MMAR_ROOT/config/dataset_0.json \
    epochs=5000 \
    learning_rate=0.0003 \
    num_training_epoch_per_valid=10 \
    multi_gpu=true
```

Note that you must set `DATA_ROOT` in the `environment.json` file to the directory containing training data. In our case, it was as follows:

```
{
        "DATA_ROOT": ""/mnt/mount_0/dataset/ixi_hippocampus/,
        "DATASET_JSON": "/mnt/mount_0/dataset/ixi_hippocampus/dataset.json",
        "PROCESSING_TASK": "segmentation",
        "MMAR_CKPT_DIR": "models",
        "MMAR_EVAL_OUTPUT_PATH": "eval",
        "PRETRAIN_WEIGHTS_FILE": "/var/tmp/resnet50_weights_tf_dim_ordering_tf_kernels.h5"
}
```

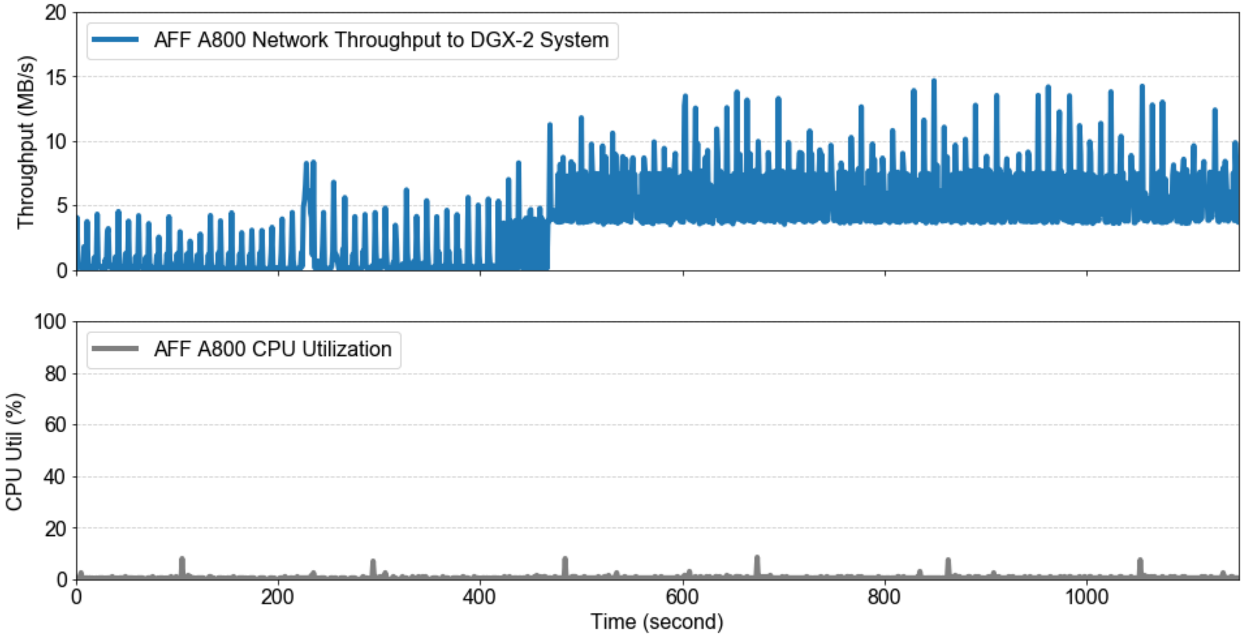
Moreover, the NVIDIA Clara Train SDK v2.0 supports automatic mixed precision (AMP) deep-neural-network training. You can elect to train with half precision while maintaining network accuracy comparable to the level achieved with single precision. The technique of using both single-precision and half-precision representations simultaneously is referred to as the mixed precision technique. Training with AMP can reduce memory requirements and speed some models up to 3X. To enable AMP, set the `use_amp` variable in `config.json` to `true`:

```
{
       "epochs": 5000,
       "num_training_epoch_per_valid": 20,
       "learning_rate": 1e-5,
       "use_amp": true,
       ...
```

Note that the NVIDIA Clara Train SDK v2.0 allows you to use the Smart Cache technique to keep data in memory for re-use and reduce demand on I/O and CPU resources. For more information on how Smart Cache works, including examples and analysis, see the Clara documentation.

Figure 5 illustrates the network throughput and CPU utilization of the AFF A800 for a single DGX-2 system with AMP enabled. This graph shows a throughput spike of less than 10MBps at the beginning, and then a sustained throughput of less than 15MBps for the remainder of the training run. This observation is caused by the fact that individual MR image slices are less than 102KB in size. Moreover, as is shown in the same graph, the peak CPU utilization of the AFF A800 is below 8.5%, indicating that significant headroom remains for additional DGX-2 systems.

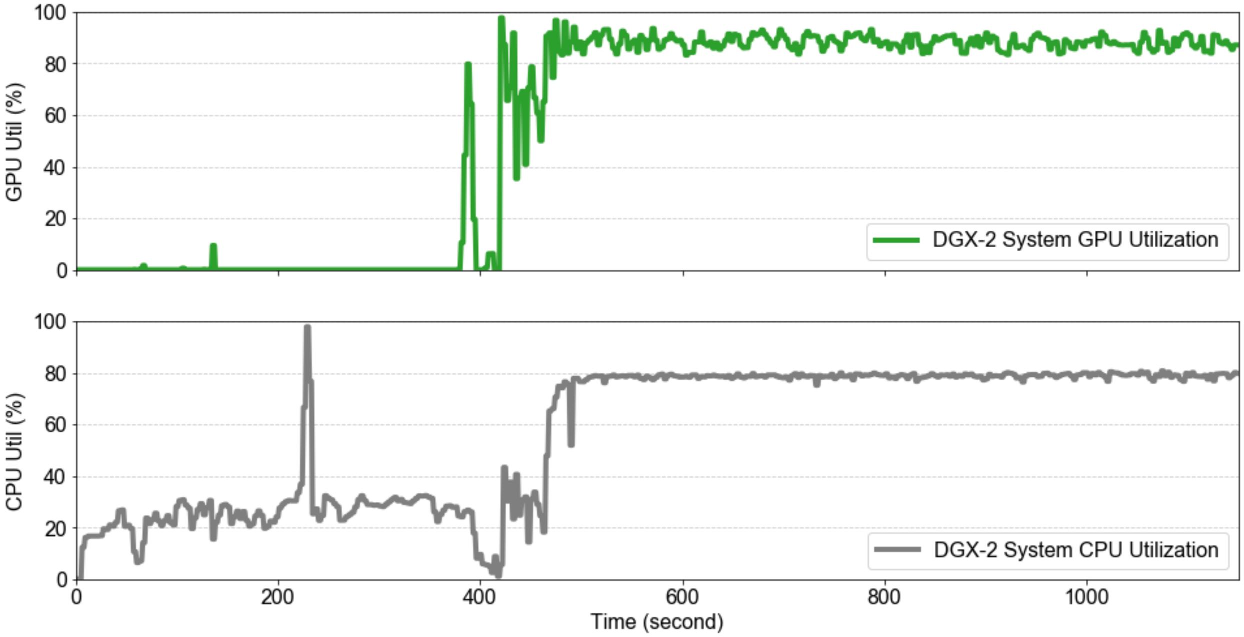
Figure 5) AFF A800 network throughput and CPU utilization for hippocampus segmentation model training.



To better demonstrate what happened during training, Figure 6 shows the CPU and GPU utilization of a single DGX-2 system. GPUs on the DGX-2 system were almost fully utilized during the model training epochs, while CPU usage stayed around 80%.

Note: Storage throughput and CPU and GPU usage without AMP were nearly identical to the data presented in Figure 5 and Figure 6. Only the total training time was affected.

Figure 6) DGX-2 system CPU and GPU utilization for hippocampus segmentation model training.



5 Solution Sizing Guidance

As validated in NVA-1135, many AI training workloads require a storage read throughput of roughly 5GBps per DGX-2 system. However, as shown in the testing above, the image segmentation model training phase only requires 5–10MBps throughput from the storage system to saturate a single DGX-2 system due to the small MR image sizes. Synthetic performance benchmarks performed in NVA-1121 show that the NetApp AFF A800 storage system delivers up to 25GBps of read performance. Therefore, based on the results described in Section 4, the AFF A800 storage system can support as many as ten DGX-2 systems with this specific training workload and still have headroom for data preparation or other tasks.

The image data used in this validation is not conducive to storage efficiency techniques such as deduplication and compression, so customers can typically expect very little savings from those features. NetApp AFF A800 systems support a variety of SSD capacity options ranging from 100TB to 800TB per AFF A800 storage system. Customers can choose the SSD size that meets their capacity requirements without any effect on performance.

For organizations that require larger numbers of DGX-2 systems, NetApp ONTAP supports storage clusters of up to 24 nodes, enabling linear scaling of capacity and performance as DGX-2 systems are added to the environment. Please consult with a NetApp technical representative about detailed sizing for specific workload requirements.

6 Conclusion

AI solutions demand high accuracy to train and evaluate model fitness. Accuracy is particularly important for assessing and predicting medical conditions in the healthcare industry. Convolutional neural networks (CNNs), a category of DL, have proven to be very effective for image recognition and classification applicable to radiology and diagnostics. Therefore, this type of neural network has seen a significant increase in adoption in recent years.

The ONTAP AI reference architecture is an optimized environment for the development of DL models for medical imaging, such as DNN and CNN, and many other healthcare use cases. With the compute power of the NVIDIA DGX-2 system and the data management capabilities of NetApp ONTAP, ONTAP AI enables a full range of data pipelines that spans the edge, the core, and the cloud.

The models and datasets tested in this solution show that ONTAP AI can easily support the workload requirements for model training with MRI datasets, helping data scientists train models to reach higher accuracy and reduce their time to value. This reference architecture enables complex operations in a shorter computation time for training DL models, adding value to healthcare businesses.

Acknowledgments

The authors gratefully acknowledge the contributions that were made to this technical report by our esteemed colleagues from NVIDIA: Jacci Cenci, Darrin Johnson, and Robert Sohigian. The authors would also like to acknowledge the contributions of key NetApp team members: Santosh Rao, David Arnette, Michael Oglesby, Erik Mulder, and Mike McNamara.

Our sincere appreciation and thanks go to all these individuals, who provided insight and expertise that greatly assisted in the creation of this paper.

Where to Find Additional Information

To learn more about the information that is described in this document, see the following resources:

- NVIDIA DGX-2 systems
  - NVIDIA DGX-2 systems
    https://www.nvidia.com/en-us/data-center/dgx-2/
  - NVIDIA Tesla V100 Tensor core GPU
    https://www.nvidia.com/en-us/data-center/tesla-v100/
  - NVIDIA GPU Cloud
    https://www.nvidia.com/en-us/gpu-cloud/
- NVIDIA Clara Train SDK
  - AI-assisted annotation
    https://docs.nvidia.com/clara/tlt-mi/clara-train-sdk-v2.0/aiaa/index.html
  - Transfer learning: multi-GPU training
    https://docs.nvidia.com/clara/tlt-mi/clara-train-sdk-v2.0/nvmidl/model.html#multi-gpu-training
  - Bring your own components
    https://docs.nvidia.com/clara/tlt-mi/clara-train-sdk-v2.0/nvmidl/byom.html
- NetApp AFF systems
  - AFF datasheet
    https://www.netapp.com/us/media/ds-3582.pdf
  - NetApp FlashAdvantage for AFF
    https://www.netapp.com/us/media/ds-3733.pdf
  - ONTAP 9.x documentation
    http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286
  - NetApp ONTAP FlexGroup technical report
    https://www.netapp.com/us/media/tr-4557.pdf
- NetApp ONTAP AI
  - ONTAP AI with DGX-1 and Cisco Networking Design Guide
    https://www.netapp.com/us/media/nva-1121-design.pdf
  - ONTAP AI with DGX-1 and Cisco Networking Deployment Guide
    https://www.netapp.com/us/media/nva-1121-deploy.pdf
  - ONTAP AI with DGX-1 and Mellanox Networking Design Guide
    http://www.netapp.com/us/media/nva-1138-design.pdf
  - ONTAP AI with DGX-2 Design Guide
    https://www.netapp.com/us/media/nva-1135-design.pdf
- ONTAP AI networking
  - Cisco Nexus 3232C series switches
    https://www.cisco.com/c/en/us/products/switches/nexus-3232c-switch/index.html
  - Mellanox Spectrum 2000-series switches
    http://www.mellanox.com/page/products_dyn?product_family=251&mtag=sn2000
- Machine learning frameworks and tools
  - TensorFlow: an open-source machine learning framework for everyone
    https://www.tensorflow.org/
  - Enabling GPUs in the Container Runtime Ecosystem
    https://devblogs.nvidia.com/gpu-containers-runtime/
- Datasets and benchmarks

- IXI brain imaging dataset
  http://brain-development.org/ixi-dataset/
- MITK Workbench
  http://www.mitk.org/download/releases/MITK-2018.04.2/Nvidia/

Version History

Version	Date	Document Version History
Version 1.0	December 2019	Initial release
Version 1.1	March 2020	Update following Clara Train SDK v2.0 release

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b).

Trademark Information

TR-4811-0220