White Paper

# NetApp ONTAP AI and allegro.ai

## Powering the Deep Learning Platform

Faiz Abidi, NetApp
Erez Schneider and Gregory Axler, allegro.ai
March 2019 | WP-7300

In partnership with

allegro.ai

## Abstract

With huge amounts of data being generated continuously, artificial intelligence (AI) has grown exponentially in the past few years and has taken the world by storm. But it becomes a challenge when you must implement and run deep learning (DL) technology repeatedly and at scale. To help you meet this challenge, allegro.ai's sophisticated and robust platform is specifically designed for this paradigm. Other effective components are the innovative features of NetApp® ONTAP® AI, which simplify overall deployment and provide high throughput by using NetApp AFF all-flash storage systems and NVIDIA graphics processing units (GPUs). By combining the allegro.ai and NetApp technologies, it has never been easier for you to create DL models and to train them. This paper briefly discusses allegro.ai's platform, ONTAP AI features, and how a combination of the two technologies can help you run your DL models more efficiently while extracting maximum performance from your GPUs.

NetApp®

## TABLE OF CONTENTS



## LIST OF TABLES

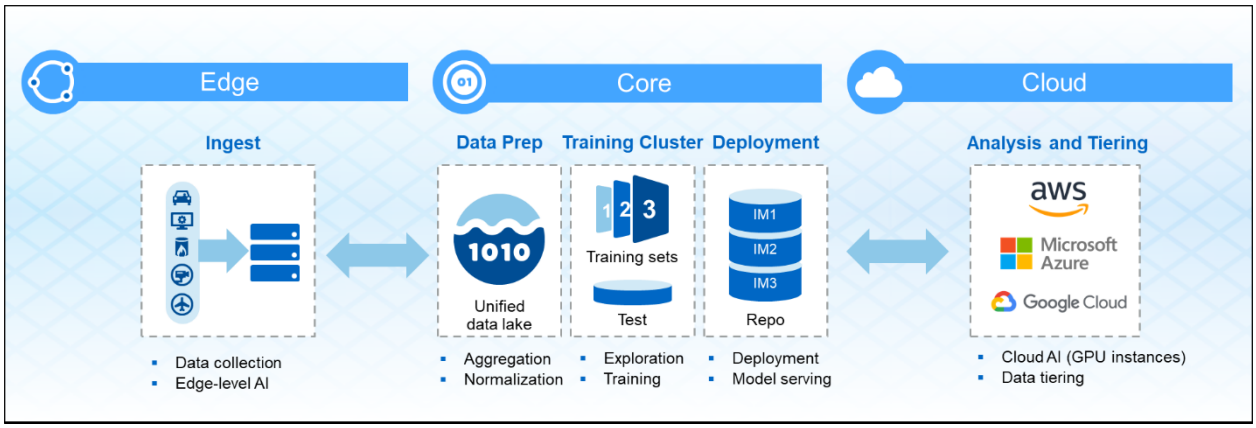


## LIST OF FIGURES

# 1 Introduction

Deep learning (DL) is the engine that enables you to detect fraud, to improve customer relationships, to optimize your supply chain, and to deliver innovative products and services in an increasingly competitive marketplace. The performance and accuracy of DL models are significantly improved by increasing the size and complexity of the neural network as well as the amount and quality of data that is used to train the models.

**Figure 1) Edge-to-core-to-cloud data pipeline.**




Given the massive datasets in a neural network, it is critical to architect an infrastructure that gives you the flexibility to deploy across environments. At a high level, an end-to-end DL deployment consists of three stages through which the data travels: the edge (data ingest), the core (training clusters and a data lake), and the cloud (archiving, tiering, and development and test). This DL deployment approach is typical in applications such as the Internet of Things (IoT) for which data spans all three realms of the data pipeline.

Figure 1 presents an overview of the components in each of the three realms:

- **Data ingest.** Data ingestion usually occurs at the edge by capturing, for example, data streaming from autonomous cars or point-of-sale (POS) devices. Depending on the use case, an IT infrastructure might be needed at or near the ingestion point. For instance, a retailer might need a small footprint in each store that consolidates data from multiple devices.
- **Data preparation.** Preprocessing is necessary to normalize and to cleanse the data before training. Preprocessing takes place in a data lake, possibly in the cloud as an Amazon S3 tier or in on-premises storage systems such as a file store or an object store.
- **Training.** For the critical training phase of DL, data is typically copied from the data lake into the training cluster at regular intervals. The servers that are used in this phase use GPUs to parallelize computations, creating a tremendous appetite for data. To maintain high GPU utilization, it is critical for your system to meet the raw I/O bandwidth needs.
- **Inference (deployment).** The trained models are tested and deployed into production. Alternatively, the models can be fed back to the data lake to further adjust their input weights, or in IoT applications, the models can be deployed to smart-edge devices.
- **Archiving and tiering.** Cold data from past iterations might be saved indefinitely. Many AI teams prefer to archive cold data to object storage in either a private or a public cloud.

Depending on the application, DL models work with large amounts of different types of data (both structured and unstructured). This difference imposes a varied set of requirements on the underlying storage system, both in terms of size of the data that is being stored and the number of files in the dataset. Some of the high-level storage requirements include:

- The ability to store and to retrieve millions of files concurrently
- Storage and retrieval of diverse data objects, such as images, audio, video, and time-series data
- Delivery of high parallel performance at low latencies to meet the GPU processing speeds
- Seamless data management and data services that span the edge, the core, and the cloud

Combined with superior cloud integration and the software-defined capabilities of NetApp ONTAP, AFF systems support a full range of data pipelines that spans the edge, the core, and the cloud for DL.

# 2 ONTAP AI Overview

The NetApp ONTAP AI proven architecture, powered by NVIDIA DGX supercomputers and NetApp cloud-connected storage, has been developed and verified by NetApp and NVIDIA. It provides a prescriptive architecture that enables your organization to:

- Eliminate design complexities.
- Independently scale compute and storage.
- Start small and scale seamlessly.
- Choose from a range of storage options for various performance and cost points.

ONTAP AI integrates NVIDIA DGX-1 servers with NVIDIA Tesla V100 GPUs and a NetApp AFF A800 system with state-of-the-art networking. ONTAP AI simplifies AI deployments by eliminating design complexity and guesswork. Your enterprise can start small and grow nondisruptively while intelligently managing data from the edge to the core to the cloud and back.

**Figure 2) ONTAP AI solution rack-scale architecture.**

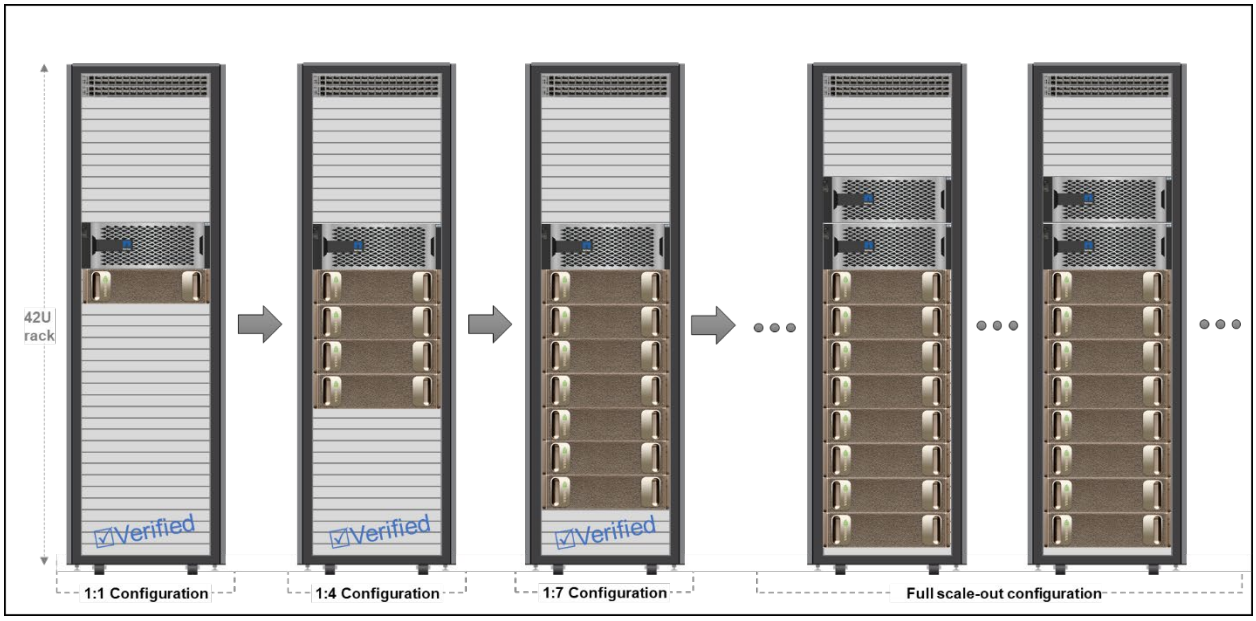



Figure 2 shows the scalability of the ONTAP AI solution. The AFF A800 system has been verified with seven DGX-1 servers and has demonstrated sufficient performance headroom to support even more DGX-1 servers without affecting storage throughput or latency. By adding more network switches and storage controller pairs to the ONTAP AI cluster, the solution can scale to multiple racks to deliver extremely high throughput and to accelerate training and inferencing. This approach gives you the flexibility to alter the ratio of compute to storage independently according to the size of your data lake, your DL models, and the performance metrics that you need.

The number of DGX-1 servers and AFF systems that you can place in a rack depends on the power and cooling specifications of the rack that you use. Final placement of your systems is subject to computational fluid dynamics analysis, airflow management, and data center design.

# 3 allegro.ai Overview

Advanced computer vision solutions have been developed through an area of AI known as DL. With these solutions, the vision of autonomous vehicles, new medical image analysis capabilities, and many other use cases across a wide range of business and product domains are turning into reality. But the challenges of this new paradigm of product development are great, and they span both data science and software engineering.

allegro.ai engineers the science of DL by providing automation, collaboration, and scale. Your engineering and product leadership get the visibility and the control that they need, while your data scientists get to focus their time on research and creative output. The result is that your organization achieves higher-quality products, faster time to market, increasing returns with scale, and significantly lower costs.

allegro.ai provides an end-to-end platform and a suite of products that you can use modularly but that create additional benefits when you use them together. Figure 3 presents an overview and Table 1 lists the feature highlights.

**Figure 3) allegro.ai computer vision DL product suite and platform.**

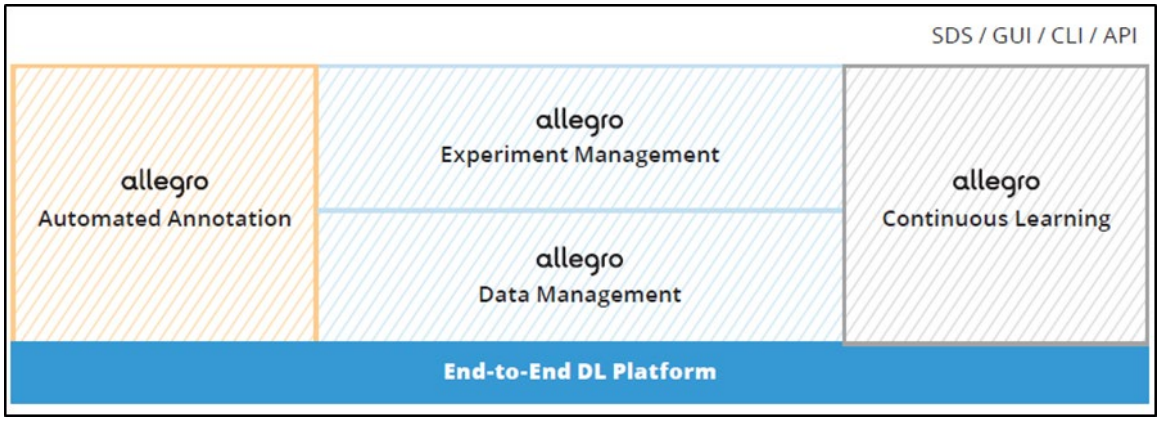



**Table 1) allegro.ai product suite and platform.**

| allegro.ai Product | Feature Highlights |
|---|---|
| Automated Annotation | • Annotation automation through integration of labeling into a core training pipeline<br>• Visual tools for annotation that support bounding boxes, polygons, segmentation, and pixel-level segmentation |
| Experiment Management | • Innovative abstractions of models and code to enable sophisticated experimentation and model building with zero programming<br>• Plug-and-play model deployment |

| allegro.ai Product | Feature Highlights |
|---|---|
| Experiment Management + Data Management | Sold together with Experiment Management, with added features such as:<br>• Powerful tools and SQL-like query capabilities to turn your fixed datasets into dynamic selection functions that you can manipulate<br>• Storage and processing virtualization for zero data move and optimized data training pipelines |
| Continuous Learning | • Postdeployment feedback loop and edge-device data integration to keep improving customer models<br>• Discrete edge-device optimization<br>• Federated learning<br>• Distributed learning |
| End-to-End DL Platform | • Total privacy of proprietary company data and models<br>• Dataset versioning that supports auditing, provenance, and reproducibility<br>• Virtualization, scalability, and distribution building blocks for all resources that are required to support DL products<br>• DevOps for DL, including scheduling and automated machine learning (AutoML)<br>• Team collaboration and management layers |

# 4 Solution Validation

To validate the allegro.ai software on NetApp ONTAP AI, we used the hardware and software that are listed in Table 2 and in Table 3, respectively.

**Table 2) Hardware details.**

| Hardware | Notes |
|---|---|
| One NVIDIA DGX-1 server | 8 Tesla V100-SMX2 GPUs, each with 32GB memory |
| One NetApp AFF A800 system | 1 high-availability (HA) pair, includes 48 x 1.92TB NVMe solid-state drives (SSDs) |
| One Cisco Nexus 3232C Switch | 100Gb Ethernet switch |

**Table 3) Software details.**

| Software | Version |
|---|---|
| Ubuntu OS | 16.04.5 LTS |
| NetApp ONTAP | 9.4 |
| Cisco NX-OS switch firmware | 7.0(3)I6(1) |
| allegro.ai | 2.1.0 |

## 4.1 Semantic Segmentation by Using Images

For the purpose of validation testing, we ran a semantic segmentation experiment on a collage that consisted of 1,300 images, and each image was 76MB in size. These images were generated from a publicly available COCO dataset that was stored on an NFS mounted volume from an AFF A800 system.

**Note:** For more information, see COCO dataset.

The goal of semantic segmentation is to label each pixel of an image with a corresponding class of what is being represented. Unlike object detection in images, the expected output in semantic segmentation is not just labels and bounding-box parameters. The output itself is a high-resolution image (typically of the same size as the input image) in which each pixel is classified to a particular class. Thus, it is a pixel-level image classification.

**Note:** For more information, see Understanding Semantic Segmentation with UNET.

Figure 4 and Figure 5 show a partial collage and part of the results, respectively, that we obtained from the semantic segmentation experiment. We ran 2,000 iterations in the experiment.

**Figure 4) Collage that was created for semantic segmentation.**

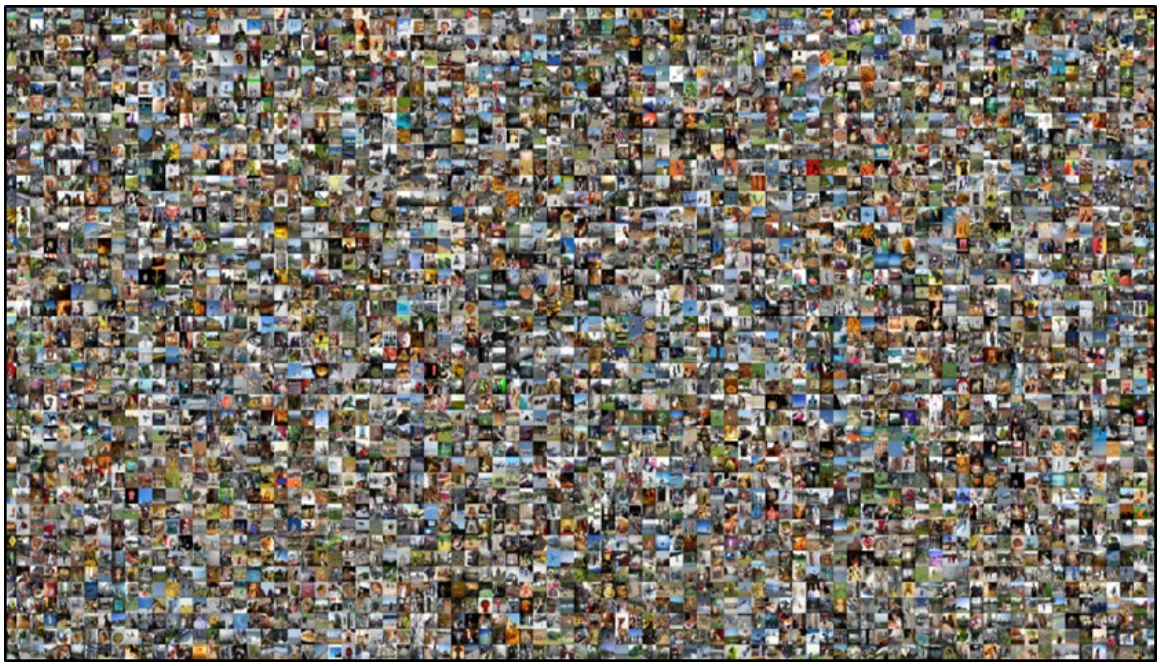


Figure 5) Sample run of images that were created.

## 4.2 Semantic Segmentation by Using Large Video Files

In our second experiment to validate our joint solution, we ran a semantic segmentation algorithm (U-Net) on a large video dataset. This dataset consisted of videos from 12 high-definition security cameras, with a total size of approximately 50GB (approximately 8.6 million frames in total).

For more information, see SAIVT-BuildingMonitoring.

To train a neural network by using video data, you must select random frames from the videos. You can select random frames in two ways:

- **Keep videos compressed and decode the necessary sequence.** This process is efficient in terms of storage that is used, but it is computationally intensive and might slow down training (because each frame must be decoded).
- **Decode the videos to images of frames and train by using image data.** This process enables fast training and increased utilization of GPU. Because decoded video might increase by a factor of 10 or more, this approach is demanding in terms of storage.

Because the first approach is CPU intensive, it might slow down the training. We tested both approaches. We called the first approach *Decoded*, and we called the second approach *Cached*. For the Cached approach, all the decoded images were stored on ONTAP AI. This approach is not possible for large datasets without the use of ONTAP AI.

Figure 6 shows a sample run of our experiment that used large video files where videos were decoded to images of frames (second approach). In section 5, we discuss the advantages of running allegro.ai software on top of ONTAP AI.

Figure 6) Sample run of semantic segmentation by using large video files.

# 5 Why Should You Use allegro.ai on ONTAP AI?

We ran two validation experiments with allegro.ai software running on top of NetApp ONTAP AI. In this section, we discuss some of the advantages of the allegro.ai and NetApp joint solution.

## 5.1 Fast Time to Completion

The time that it takes to train DL models is an important factor when you are building AI pipelines. If a model takes too long to be trained, you lose valuable time before you can put it into production. To speed up training time, you can add more GPUs. Often, however, the data I/O becomes the bottleneck, causing significant delays in the overall time to completion.

ONTAP AI provides high-speed data I/O and throughput. Because all the data that we used in our DL experiments was stored on a high-performing ONTAP AI file system, the data I/O time decreased significantly, making the overall model training fast.

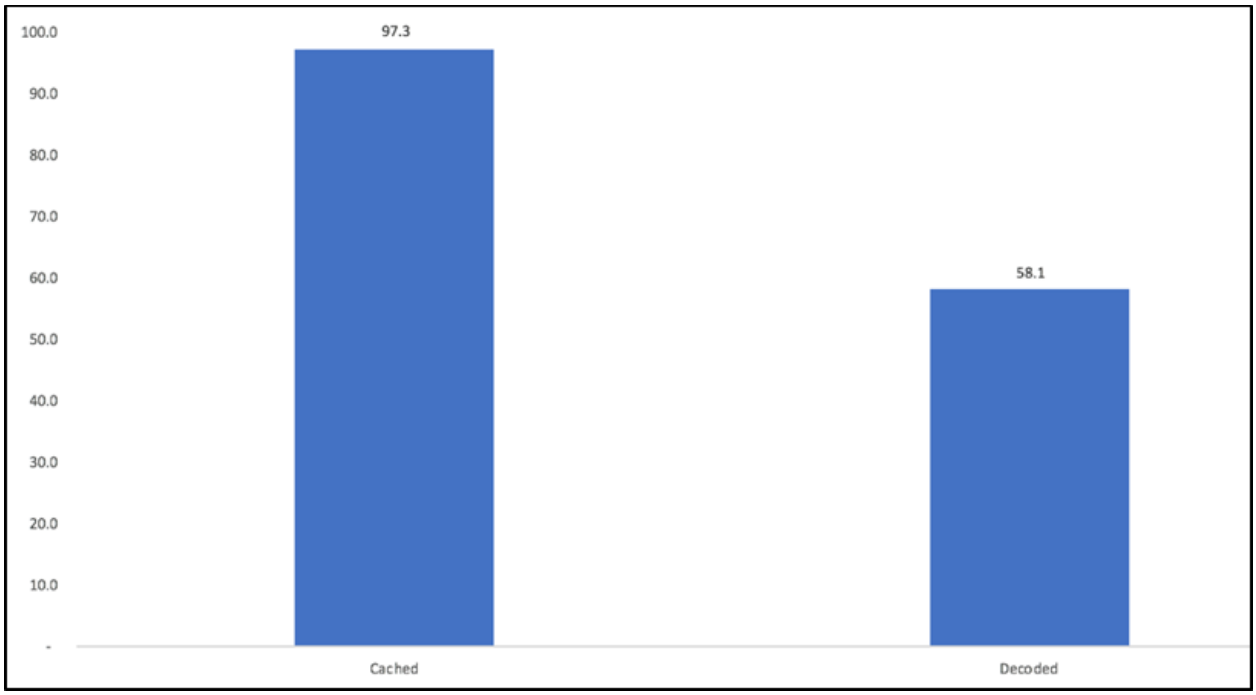
## 5.2 High GPU Utilization

Typically, when you run DL experiments with big data, you spend a lot of time on data access and on writing metadata. And while that data I/O occurs, GPUs do not have much work to do and just sit idle. That GPU idle time is even more significant when you have a big GPU cluster that comprises several servers, with each server containing multiple GPUs.

When you store data on ONTAP AI, which provides extremely high data I/O and throughput speeds, you essentially cut down on GPU idle time by keeping GPUs engaged most of the time. Your models train faster, and you can train more models in a given time.

Figure 7 shows the GPU utilization percentage that we obtained by running the experiment with large video files in our laboratory. By running allegro.ai on ONTAP AI, the GPU utilization reached as high as 97.3% on average during the caching-based experiment, and GPU utilization reached 58.1% on average during the decoding-based experiment.

**Figure 7) Average GPU utilization percentage during caching and decoding while running the video experiment.**




### 5.3 Large Cache Size

Having a large cache size also plays an important role in the overall time to completion when you run DL models. Because ONTAP AI provides large pools of flash storage, you can use it to cache your data. In our semantic segmentation experiment that used large video files, we cached our data on ONTAP AI. allegro.ai software decoded the videos to images of frames and used that cached data for training. Without extensive and fast data storage like ONTAP AI provides, our experiment would not have been possible.

## 6 Conclusion

More than the code, it is the data that is of primary importance in a DL system. A neural network performs well only if it is thoroughly trained with a big enough dataset. But big data comes with its own set of challenges, such as data that is skewed, unstructured, and unlabeled. Also, lack of a version control system and lack of QA add to the set of quality issues that you can encounter while you build an effective training model. Other problems, such as increased time to fetch the data for training and lower GPU utilization, can contribute to the overall poor performance of the DL model. Allegro.ai software on top of NetApp ONTAP AI can help you solve these problems. You get data management, automated annotation, training and model management, and continuous learning. And because the data and the cache are stored on ONTAP AI, the entire DL pipeline accelerates, which in turn increases GPU utilization.

## Acknowledgments

We would like to thank the following people for their help and for their contributions to this paper:

- David Arnette, Technical Marketing Engineer, NetApp

- Sundar Ranganathan, Senior Product Manager, NetApp
- Amit Borulkar, Technical Marketing Engineer, NetApp
- Sung-Han Lin, Performance Analyst, NetApp
- Santosh Rao, Senior Technical Director, NetApp
- Karthikeyan Nagalingam, Principal Architect (Big Data Analytics), NetApp
- Moses Guttmann, CTO, allegro.ai
- Gil Westrich, Vice President of R&D, allegro.ai
- Dan Malowany, Head of DL Research, allegro.ai
- Anna Giaconia, Copy Editor, NetApp
- Heather Kennedy, Technical Editor, NetApp

## Where to Find Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- NetApp ONTAP AI, Powered by NVIDIA
  https://www.netapp.com/us/media/nva-1121-design.pdf
- NetApp ONTAP AI Solution Brief
  https://www.netapp.com/us/media/sb-3939.pdf
- COCO dataset
  http://cocodataset.org/#home
- SSD object detection
  https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06
- allegro.ai
  https://allegro.ai/
- SAIVT-BuildingMonitoring
  https://research.qut.edu.au/saivt/databases/saivt-buildingmonitoring/
- Understanding Semantic Segmentation with UNET
  https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47

## Version History

| Version | Date | Document Version History |
|---|---|---|
| Version 1.0 | March 2019 | Initial release, including validation testing with allegro.ai. |

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**



**Trademark Information**



WP-7300-0319