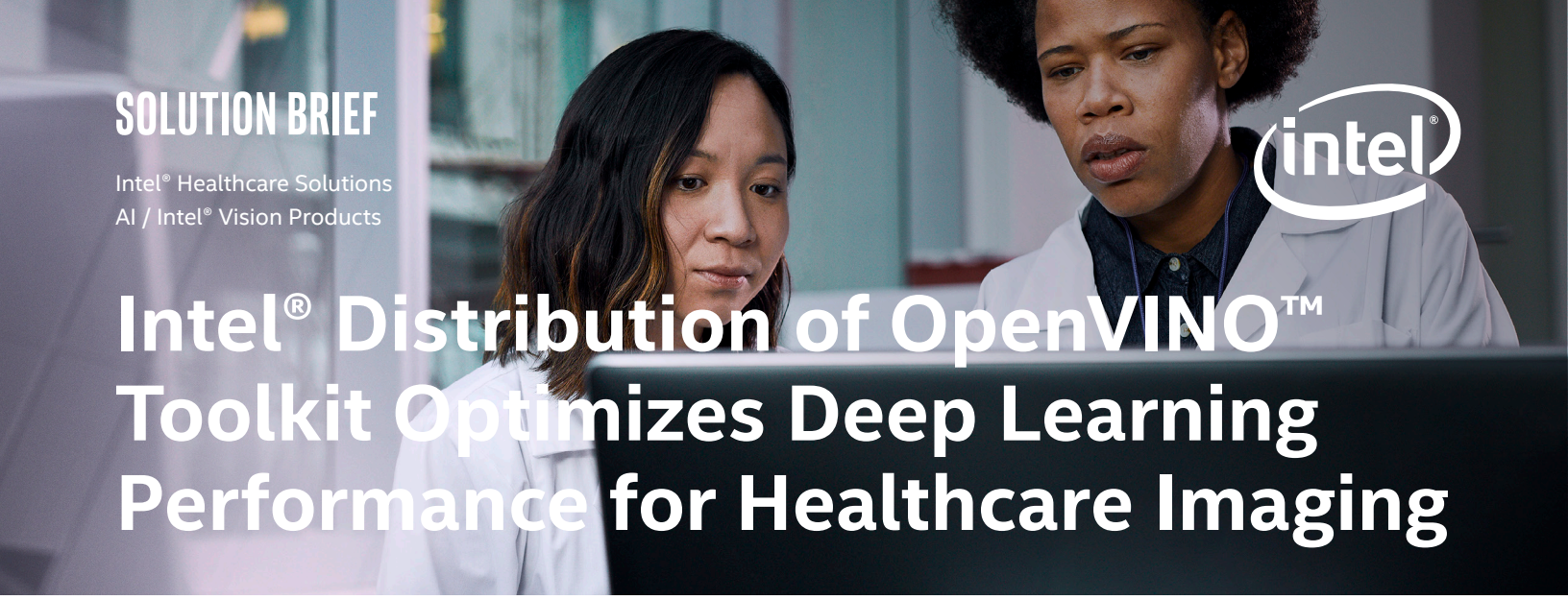


SOLUTION BRIEF

Intel® Healthcare Solutions
AI / Intel® Vision Products

Intel® Distribution of OpenVINO™ Toolkit Optimizes Deep Learning Performance for Healthcare Imaging

The Intel® Distribution of OpenVINO™ toolkit helps developers and data scientists accelerate development of computer vision and deep learning applications, and deliver the power of AI to clinical diagnostic scanning and other healthcare workflows—without adding hardware cost and complexity.

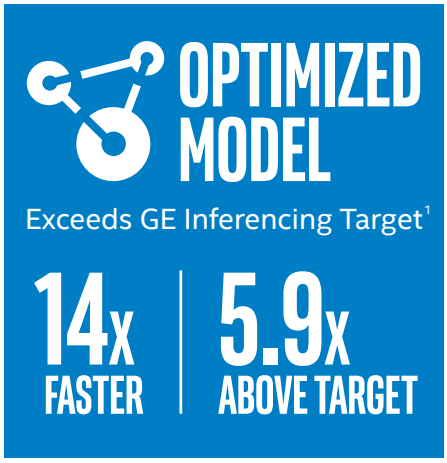


Table of Contents

Executive Summary

Deep learning and other forms of artificial intelligence (AI) offer exciting potential to streamline medical imaging workflows, enhance image quality, and increase the research value of imaging data. To deliver on the promise of AI-enhanced medical imaging, developers and other innovators must deploy their solutions on flexible platforms that provide high performance and scalability for deep learning innovations without driving up costs.

The Intel AI portfolio offers unprecedented choice in hardware that is optimized for the entire data workflow—from edge to cloud. In addition, Intel enables developers with optimizations for popular frameworks and abstractions on Intel® hardware, from CPUs to VPUs and FPGAs, across training and inference workloads. Intel also offers specialized software options to boost and support AI development like the Intel® Distribution of OpenVINO™ toolkit, which includes the Intel® Math Kernel Library for Deep Neural Networks (Intel® MKL-DNN) to allow for direct optimization of algorithms across Intel® processors. These technologies provide an easy way for innovators to optimally deploy and integrate their deep learning models on various Intel® architectures from a variety of frameworks and training platforms. Developers can leverage the existing Intel architecture in their datacenter and edge devices to deploy AI innovations without adding costs and complexity.

Intel and GE Healthcare explored the use of Intel® AI technologies with one of GE's deep learning image-classification solutions. They found that optimizing the trained GE solution with the Intel Distribution of OpenVINO toolkit improved throughput an average of 14 times over a baseline version of the solution and exceeded GE's throughput goals by almost six times.[1] These findings show a path to practical AI deployment for next-generation diagnostic scanners and a new era of smarter medical imaging.

Transforming Healthcare with Deep Learning Imaging

Medical images are valuable in diagnosing a wide range of health issues, as well as planning treatment and assessing its results. Aggregated with other sources of healthcare and demographic information, imaging data can also lead to novel insights that inspire next-generation treatment breakthroughs.

Deep learning and other forms of AI offer exciting potential to improve medical imaging workflows and enhance medical imaging quality. Deep learning is a branch of machine learning in which developers create mathematical and neural network models and use vast amounts of data to "train" them to perform tasks such as recognizing and classifying medical images. Once a trained model is sufficiently accurate, it can be deployed with other algorithms as an "inference engine," to evaluate and categorize real-world inputs from medical imaging devices like X-ray machines and CT scanners.

For medical imaging AI solutions, the deployment architecture must deliver high inferencing throughput that keeps pace with busy radiology workflows but doesn't restrict flexibility or add needless complexity and costs to the deployment environment.

Solution Overview: Deploying Deep Learning Inference

In addition to its powerful processors, memory, and storage solutions, Intel offers tools for optimized inferencing on flexible, cost-effective Intel architecture.

The Intel Distribution of OpenVINO toolkit is a free set of tools that lets developers optimize deep learning models for faster execution on Intel processors and accelerators. The toolkit imports trained models from Caffe*, MXNet*, TensorFlow*, ONNX*, and other popular deep learning frameworks, regardless of the hardware platforms used to train the models. Developers can quickly integrate various trained neural network models with application logic using a unified application programming interface (API). The toolkit maximizes inference performance by reducing the solution's overall footprint and optimizing performance for Intel hardware.

The toolkit also enhances models for improved execution, storage, and transmission, and it enables seamless integration with application logic. Developers can also test and validate their models on the target hardware environment to confirm accuracy and performance. The result is an embedded-friendly inferencing solution with a small footprint and excellent performance for high-throughput deployment on industry-standard technologies.

The Intel Distribution of OpenVINO toolkit includes the Intel MKL-DNN, a high-performance library designed to accelerate neural network primitives, increase application performance, and reduce development time. Intel MKL-DNN is also available as a standalone, open source package.

Putting Intel's Inferencing Performance to the Test

GE Healthcare, a leading provider of medical imaging equipment and other healthcare technologies, worked with Intel to test inferencing performance for one of GE's deep learning solutions. Developed by a team from GE Healthcare's Computed Tomography (CT) scan division, the solution classifies and tags scanned image slices, making it easier to find relevant images and use them for research or clinical comparison.

GE's CT imaging specialists developed and trained their AI model using the Python* programming language and open source software, including the TensorFlow and Keras* libraries for deep learning. Collaborating with some of Intel's AI optimization experts, they used the Intel Distribution of OpenVINO toolkit to set up, test, and optimize their solution on existing Intel architecture-based infrastructure. Reflecting the need for cost-effective inference performance, the CT experts set a target of classifying 100 images per second using no more than four dedicated cores of an Intel® Xeon® processor.

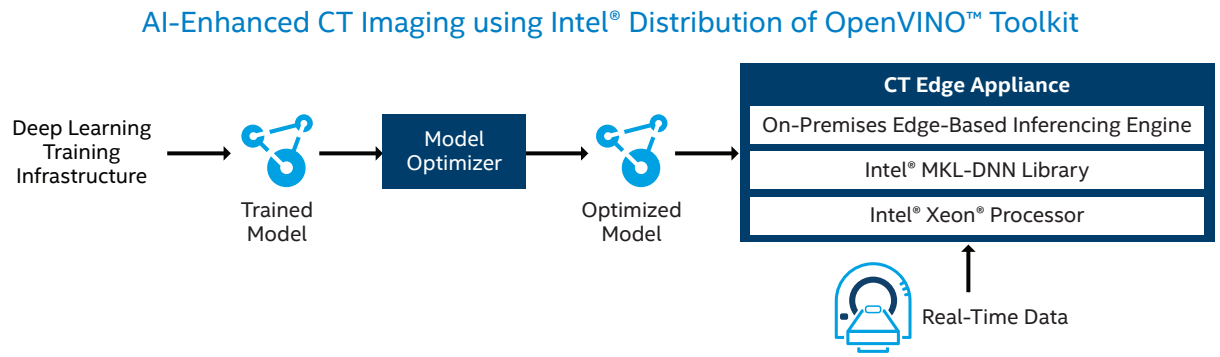


Figure 1. By optimizing models with the Intel® MKL-DNN, a neural network library within the Intel® Distribution of OpenVINO™ toolkit, GE Healthcare has been able to achieve outstanding inferencing performance at the enterprise edge in clinical and research settings.

The results were impressive. The optimized codes produced by Intel MKL-DNN, a neural network library within the Intel Distribution of OpenVINO toolkit, improved inferencing throughput an average of 14 times over the baseline TensorFlow model running on the same system, and easily met GE's performance target. In fact, a single core of the Intel® Xeon® processor E5-2650 ran nearly 150 percent faster than GE's performance target, and four cores of the processor exceeded GE Healthcare's goal by nearly six times (see Figure 2). Intel benchmarks show the new Intel® Xeon® Platinum 8180 processor delivering up to 2.4 times higher performance for a range of AI workloads compared to the previous generation.[2]

"We want to keep deployment costs down for our customers, so we need the performance and flexibility to run a whole range of AI and imaging processing tasks in a variety of clinical environments. We think using multi-purpose processors along with Intel-optimized frameworks and software tools from Intel can offer a cost-effective way to leverage AI in medical imaging in new and meaningful ways."

David Chevalier, principal engineer, GE Healthcare

Flexible Deployment

By delivering excellent inferencing performance on just a handful of CPU cores, Intel Xeon processors and the Intel Distribution of OpenVINO toolkit enable innovators such as GE Healthcare to run AI solutions at high performance on the same cost-effective edge-server infrastructure that handles local image-processing tasks such as reconstruction, registration, segmentation, and noise reduction. AI developers can take advantage of deep learning frameworks optimized for Intel hardware and the tools used to deploy their inferencing solutions on Intel architecture, regardless of the development and training environment they used.

CPU-based inferencing on Intel processors also offers innovators such as GE the flexibility to change where inferencing occurs. While medical-imaging AI deployments currently run in inferencing appliances at the on-premises server, next-generation inferencing can move to an on-premises or off-premises cloud, or into the scanner itself (Figure 3).

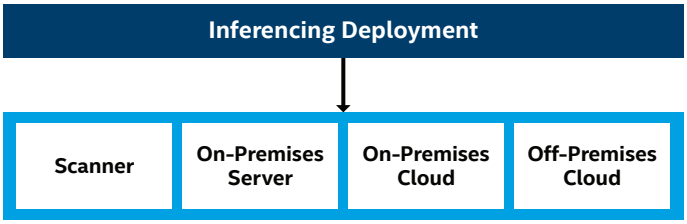


Figure 3. Running on Intel® processors gives AI innovators the flexibility to deploy inferencing capabilities in diagnostic imaging equipment, on-premises infrastructure, or secure external clouds.

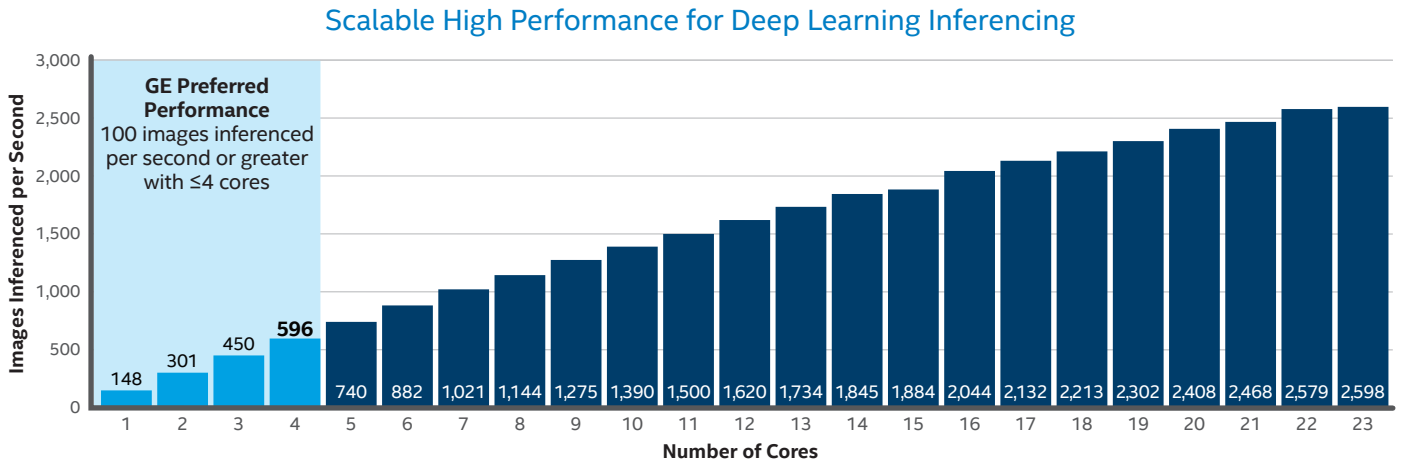


Figure 2. The Intel® Xeon® processor E5-2650 exceeded GE's inferencing performance target by nearly 6X, and demonstrated steady, scalable performance for solutions needing more cores. Performance results are based on testing as of February 2018 and may not reflect all publicly available security updates. Testing configuration: Intel® Xeon® processor E5-2650 v4 at 2.20GHz, 264 GB of memory, Intel® Solid State Drive Data Center 480 GB, and CentOS Linux* 7.4.1708 and a privacy-protected data set of 8,834 CT scan images.

Conclusion

With high-performance deep learning inferencing running on flexible Intel architecture, innovators such as GE Healthcare can deliver cost-effective AI-based medical imaging to improve image quality, diagnostic capabilities, and clinical workflows for better patient outcomes.

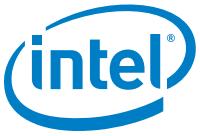
Find the solution that is right for your organization. Contact your Intel representative or visit **intel.com/healthcare**.

Learn More

You may find the following resources useful:

- Intel® Distribution of OpenVINO™ toolkit
- Intel® Math Kernel Library for Deep Neural Networks
- Framework Optimizations for Intel® Architecture
- Intel® AI Solutions for Health and Life Sciences

Solution Provided By:



Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to intel.com/benchmarks. Benchmark results were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown". Implementation of these updates may make these results inapplicable to your device or system.

[1] Testing configuration: Intel® Xeon® processor E5-2650 v4 at 2.20GHz, 264 GB of memory, Intel® Solid State Drive Data Center 480 GB, and CentOS Linux* 7.4.1708 and a privacy-protected data set of 8,834 CT scan images. Performance results are based on internal testing as of February 2018 that was validated by GE and may not reflect all publicly available security updates. No product or component can be absolutely secure.

[2] Inference throughput batch size 1, training throughput batch size 256. Platform: 2S Intel® Xeon® Platinum 8180 CPU @ 2.50GHz (28 cores), HT disabled, turbo disabled, scaling governor set to "performance" via intel_pstate driver, 384GB DDR4-2666 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86_64. SSD: Intel® SSD DC S3700 Series (800GB, 2.5in SATA 6Gb/s, 25nm, MLC). Performance measured with: Environment variables: KMP_AFFINITY='granularity=fine, compact', OMP_NUM_THREADS=56, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance. Deep Learning Frameworks: Neon: ZP/MKL_CHWN branch commit id:52bd02acb947a2adabb8a227166a7da5d9123b6d. Dummy data was used. The main.py script was used for benchmarking in mkl mode. ICC version used : 17.0.3 20170404, Intel® MKL small libraries version 2018.0.20170425. Platform: Platform: 2S Intel® Xeon® CPU E5-2699 v4 @ 2.20GHz (22 cores), HT enabled, turbo disabled, scaling governor set to "performance" via acpi-cpufreq driver, 256GB DDR4-2133 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86_64. SSD: Intel® SSD DC S3500 Series (480GB, 2.5in SATA 6Gb/s, 20nm, MLC). Performance measured with: Environment variables: KMP_AFFINITY='granularity=fine, compact,1,0', OMP_NUM_THREADS=44, CPU Freq set with cpupower frequency-set -d 2.2G -u 2.2G -g performance. Deep Learning Frameworks: Neon: ZP/MKL_CHWN branch commit id:52bd02acb947a2adabb8a227166a7da5d9123b6d. Dummy data was used. The main.py script was used for benchmarking in mkl mode. ICC version used : 17.0.3 20170404, Intel MKL small libraries version 2018.0.20170425. Performance results are based on Intel internal testing as of June 2017 and may not reflect all publicly available security updates. No product or component can be absolutely secure.

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

0219/JSTAR/KC/PDF 337177-002US