CASE STUDY

Intel® AI
Intel® Xeon® Scalable processors
OpenVINO™ toolkit

Perform AI-Driven Medical Imaging Efficiently and Cost-Effectively on Intel® CPU–Based Systems

Philips demonstrated breakthrough performance for AI inferencing of health-care workloads run on servers powered by Intel® Xeon® Scalable processors and optimized with the OpenVINO™ toolkit.

Bone-Age-Prediction Model
188X INCREASE
in Images per Second

Lung-Segmentation Model
38X INCREASE
in Images per Second

Contents

Intel teamed up with Philips to show that servers powered by Intel® Xeon® Scalable processors could be used to efficiently perform deep learning inference on patients' X-rays and computed tomography (CT) scans, without the need for accelerators. The ultimate goal for Philips is to offer artificial intelligence (AI) to its end customers without significantly increasing the cost of the customers' systems and without requiring modifications to the hardware deployed in the field.

The companies tested two healthcare use cases for deep learning inference models: one on X-rays of bones for bone-age-prediction modeling, and the other on CT scans of lungs for lung segmentation. Using the OpenVINO™ toolkit and other optimizations, along with efficient multi-core processing from Intel Xeon Scalable processors, Philips was able to achieve a speed improvement of 188.1x for the bone-age-prediction model, and a 37.7x speed improvement for the lung-segmentation model over the baseline measurements. (See Appendix A for configuration details.)

AI-Enhanced Medical Imaging

AI techniques such as object detection and segmentation offer unique possibilities to help radiologists identify issues faster and more accurately, which can translate to better prioritization of cases, better outcomes for more patients, and reduced costs for hospitals.

However, AI for medical imaging is often challenging because the information is often high-resolution and multi-dimensional. Down-sampling images to lower resolutions because of memory constraints can cause misdiagnoses, unless the biomarkers are preserved. Once an AI model is trained to acceptable levels of accuracy, it needs to be incorporated into the imaging modality architecture. Given how large radiology images typically are, it is critical to be able to process these images efficiently without slowing down radiologists' workflows or impacting the accuracy of the models.

Making the Right AI Hardware Choice

Until recently, there was one prominent hardware solution to accelerate deep learning: graphics processing units (GPUs). By design, GPUs work well with images, but they also have inherent memory constraints that data scientists have had to work around when building some models.

Today, data scientists have another option. With the introduction of Intel Xeon Scalable processors in 2017, more complex, hybrid workloads could be accelerated, including larger, memory-intensive models typically found in medical imaging. For a large subset of AI workloads, Intel and Philips found that Intel Xeon Scalable processors can better meet data scientists' needs than GPU-based systems. This enables Philips to offer AI solutions at lower costs to its customers. its customers.

Test Results: Optimizing Two Deep Learning Models for Inference

Philips is developing sophisticated deep learning models for segmenting regions of interest on medical images and for medical-image classification. This case study describes use cases for optimizing the deployment of two deep learning models developed by Philips.

Use Case 1: Bone-Age-Prediction Model

The first model takes inputs from X-ray images of human bones, such as a wrist, along with a patient's gender. The inference model then determines a predicted age from the bone, in order to help identify medical conditions that lead to bone loss. For example, if the predicted age for a younger patient is less than the actual age, the patient could be suffering from malnutrition. The trained model from Philips is based on the Xception* architecture.

Use Case 2: Lung-Segmentation Model

The second inference model identifies the lungs from a CT scan of a patient's chest, and it then creates a segmentation mask around the detected organ. The results can be used to measure size and volume of the lungs or to load organ-specific disease screening models for tuberculosis or pneumothorax detection, for example. In addition, by isolating the lung in the image, a radiologist can have a clearer anatomical view of the organ, free of distraction from other structures. The trained model from Philips is based on the popular U-Net* topology.

Test Procedure

For both inference models, engineers first took baseline measurements without optimizations. Then, various optimizations were applied, as described below.

All testing was performed on a two-socket system powered by Intel Xeon Platinum 8168 processors. Full configuration details are provided in Appendix A.

Baseline Performance Measurements

Baseline measurements using inference based on Keras* and TensorFlow* were as follows:

- Bone-age-prediction model: 1.42 images per second
- Lung-segmentation model: 1.9 images per second

Optimizing the AI Models for Deployment

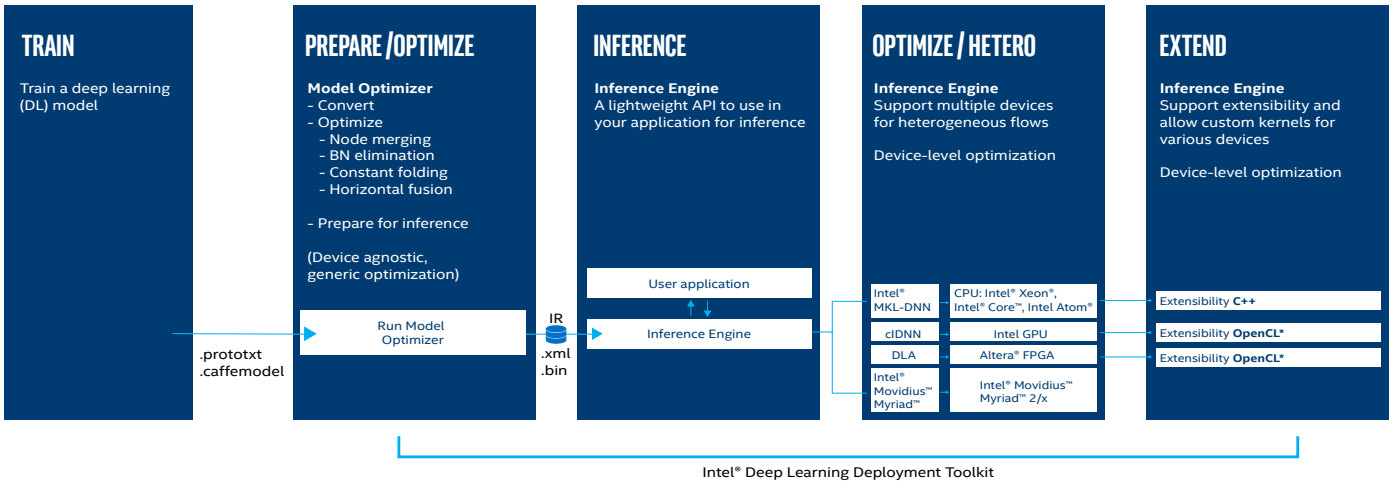
The following optimizations were performed to maximize performance for the inference models.

1. Using the OpenVINO Toolkit

Inference models for both use cases were optimized using the Intel® Deep Learning Deployment Toolkit (DLDT), which is a part of the OpenVINO toolkit. Figure 1 shows the inference workflow, from a trained deep learning model to model optimization and inference execution.

The Intel Deep Learning Deployment Toolkit contains two major components: the Model Optimizer and the Inference Engine. Philips* models—trained in Keras and TensorFlow—were first run through the Model Optimizer. The Model Optimizer performs optimizations on the neural network graphs, such as node merging, batch normalization elimination, and constant folding. The resulting output is an intermediate representation (IR) .xml file and a .bin file that contains the model weights. Model optimization is a one-time, offline process.

Next, the IR files are programmatically loaded into the Inference Engine, along with information specifying the target hardware back end, which can be any Intel Xeon processor, Intel® Core™ processor, Intel Atom® processor, an Intel® GPU, an Intel® field-programmable gate array (FPGA), or an Intel® Movidius™ Myriad™ vision processing unit (VPU).



Legend
IR = Intermediate representation
BN = Batch normalization
Intel® MKL-DNN = Intel® Math Kernel Library for Deep Neutral Networks
clDNN = Compute Library for Deep Neural Networks
DLA = Deep learning application
FPGA = Field-programmable gate array

Figure 1. The Intel® Deep Learning Deployment Toolkit (part of the OpenVINO™ toolkit) optimizes the trained model, performs inference analysis, and provides an API for applications to use to send data to the inference engine

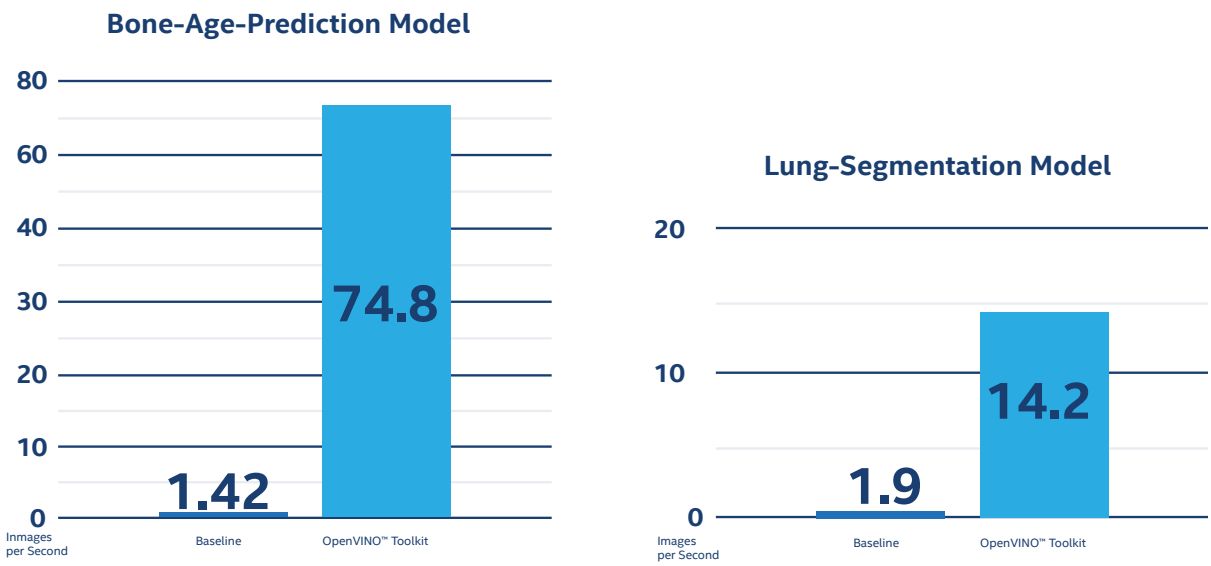


Figure 2. Inference performance increased dramatically after optimizations from the OpenVINO™ toolkit[1]

Intel has libraries for each of the hardware types that implement highly efficient deep learning kernels, such as convolutions, rectified linear unit (RELU), and others. For this Philips use case, the target back end is an Intel Xeon processor. Therefore, the library associated with CPUs (the Intel® Math Kernel Library for Deep Neural Networks [Intel® MKL-DNN]) is loaded.

The Inference Engine provides lightweight APIs in C++ and Python* that can be accessed by the custom Philips application. The application calls the APIs and inputs the image data. The Inference Engine then executes the inference and provides the results.

The baseline results improved significantly after optimizations from the OpenVINO toolkit, as shown in Figure 2.

2. Parallelize the Workload: Running Multiple Instances of the OpenVINO Toolkit

Higher throughput (images per second) can be obtained by running multiple instances of the OpenVINO toolkit on each of the sockets, instead of running just one instance of the toolkit. Each instance is bound to one or more cores, which results in better core utilization.

For the bone-age-prediction model, Intel and Philips achieved maximum throughput running 24 concurrent OpenVINO toolkit instances and binding each instance (batch size = 1) to two cores. As shown in Figure 4, this resulted in the throughput increasing from 74.8 images per second to 267.1 images per second, which is 3.6 times faster than running a single toolkit instance on all 48 cores. In addition, this optimized result is 188.1 times faster than baseline performance.

For the lung-segmentation model, Intel and Philips achieved maximum throughput running 12 instances and binding each instance (batch size = 1) to four cores. This resulted in increased throughput from 14.2 images per second to 37.0 images per second: 2.6 times faster performance than running a single OpenVINO toolkit instance on all 48 cores. These optimizations also increased performance 19.5 times over the baseline inference performance.

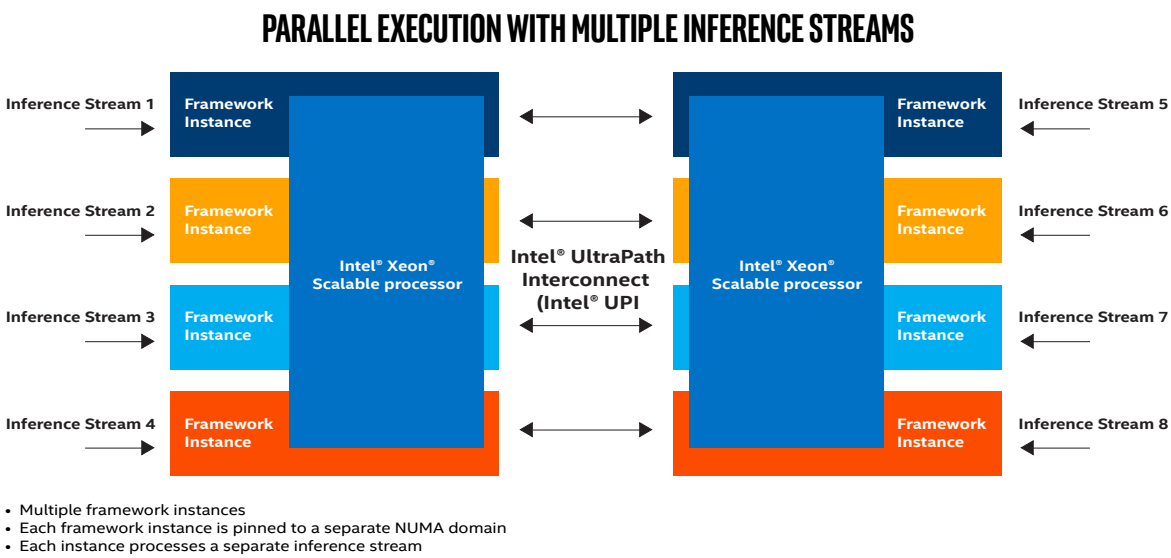


Figure 3. Sub-socket partitioning across dual-socket Intel® Xeon® platforms for multiple inference streams

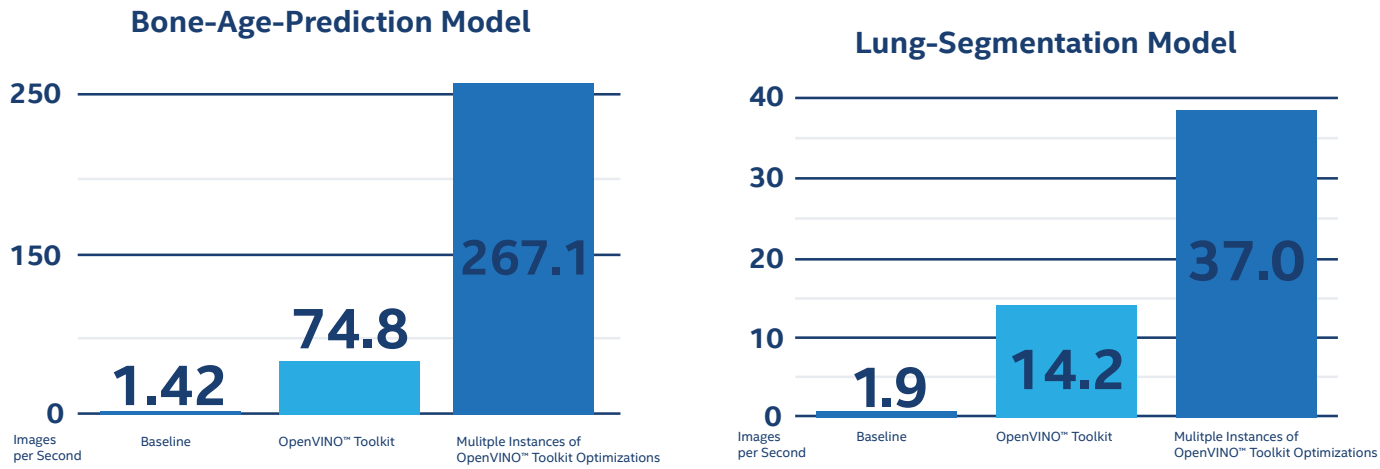


Figure 4. Parallel execution optimizations further improved inference execution performance[1]

3. Additional Optimizations

Winograd* convolutions and optimizations for resampling were applied to the lung-segmentation model (U-Net architecture). Conventional General Matrix Multiplication (GEMM)-based convolution is fast for large filters, but many state-of-the-art convolutional neural networks use small, 3 x 3 filters. The Philips topology used in the testing had several 3 x 3 convolutions that could be computed more efficiently using Winograd convolutions. With this change, Intel and Philips obtained optimal performance by increasing the number of instances to 24, binding each instance to two cores.

These optimizations were not applicable to the bone-age-prediction model.

With these techniques, lung-segmentation inference performance reached a maximum throughput of 71.7 images per second, which is 37.7x faster than the baseline inference performance (see Figure 5).

Assessing the Results

The results for both use cases surpassed expectations. The bone-age-prediction model went from an initial baseline test result of 1.42 images per second to a final tested rate of 267.1 images per second after optimizations—an increase of 188.1x. The lung-segmentation model far surpassed the target of 15 images per second by improving from a baseline of 1.9 images per second to 71.7 images per second after optimizations.

Vijayananda J., Chief Architect and Fellow, Data Science and AI at Philips HealthSuite Insights was excited to see such outstanding performance from a CPU-based system. “Intel Xeon Scalable processors appears to be the right solution for this type of AI workload. Our customers can use their existing hardware to its maximum potential, without having to complicate their infrastructure, while still aiming to achieve quality output resolution at exceptional speeds.”

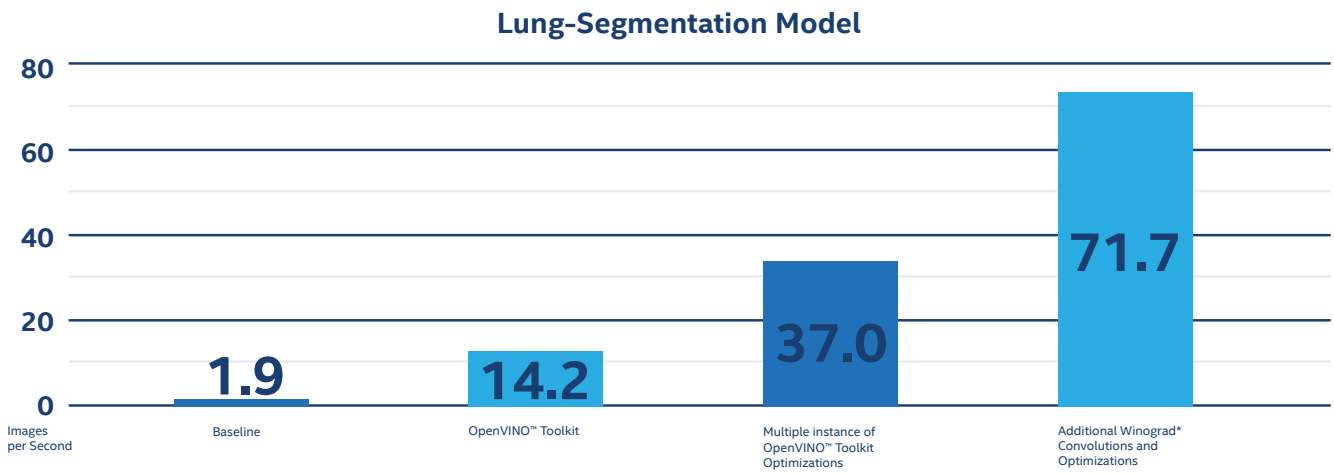


Figure 5. Additional optimizations helped boost inference execution performance for the lung-segmentation model[1]

Healthcare AI Inferencing on Affordable CPU-Based Systems

Inferencing applications in healthcare typically process workloads in small batches or in a streaming manner, which means they do not exhibit large batch sizes. CPUs are a great fit for these types of low batch or streaming applications. In particular, Intel Xeon Scalable processors offer an affordable, flexible platform for AI models—particularly in conjunction with tools like the OpenVINO toolkit, which can help deploy pre-trained models for efficiency, without sacrificing accuracy.

The imaging use cases in this study show that healthcare organizations can implement healthcare AI workloads without expensive hardware investments. And companies like Philips can offer AI algorithms for download through an online store as a way to increase revenue and differentiate themselves from growing competition.

The OpenVINO™ Toolkit Accelerates Deep Learning Deployments

This case study used the Intel® Deep Learning Deployment Toolkit—just one of several tools built into the OpenVINO toolkit. Based on convolutional neural networks (CNN), the toolkit:

- Enables CNN-based deep learning inference on the edge
- Supports heterogeneous execution across computer vision accelerators—CPU, GPU, Intel® Movidius™ Neural Compute Stick, and FPGA—using a common API
- Speeds time to market via a library of functions and pre-optimized kernels
- Includes optimized calls for OpenCV* and OpenVX*

Learn more on the **OpenVINO web site**.

Learn More

See how Philips is transforming healthcare with AI at **usa.philips.com/healthcare/innovation/artificial-intelligence**.

Learn about the full range of AI tools available for developers and explore other ways companies are using Intel® technologies to power AI at **ai.intel.com**.

Learn more about Intel Xeon Scalable processors at **intel.com/xeonscalable**.

Appendix A

Hardware configuration details:

Model Name	Intel® Xeon® Platinum 8168 processor at 2.70 GHz, Intel® Hyper-Threading Technology (Intel® HT Technology) disabled
BIOS Version	SE5C620.86B.0D.01.0010.072020182008
System Memory	192 GB, 2,666 MHz
Intel® Turbo Boost Technology	Enabled

Solid state drive (SSD) details:

ATA Device, with Non-removable Media	
Model Number	INTEL SSDSC2CW240A3

Software configuration details:

Ubuntu*	Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-29-generic x86_64*)
Keras*	2.1.1
TensorFlow*	1.2.1
OpenVINO™ Toolkit	2018 R2
Intel® Math Kernel Library	Intel Math Kernel Library for Deep Neural Networks (Intel® MKL-DNN) v0.14

Datasets:

Bone-Age-Prediction Model	299x299x3 .png images
Lung-Segmentation Model	512x512 .dcm images

[1] Results were determined with the same system configuration as shown in Appendix A. The baseline shows zero optimizations. The optimized data used the same system configuration, in conjunction with the noted optimizations.

Performance results are based on testing as of August 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.

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.

Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

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.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Philips disclaims all express and implied warranties whatsoever, including without limitation, the implied warranties of merchantability, non-infringement and fitness for any particular purpose. Further, Philips will not be liable for any direct, indirect, special, incidental, punitive, or consequential damages of any kind.

Printed in USA 0818/CVN/PRW/PDF Please Recycle 337975-001US