White Paper

Healthcare
Clinical Ultrasound Systems

FUJIFILM VisualSonics Accelerates AI for Ultrasound with Intel® Technologies

Running the Intel® Distribution of OpenVINO™ Toolkit on Intel® Core™ processors results in a huge performance improvement in ultra-high frequency, AI-enabled, ultrasound applications.

Authors

Christopher A. White
Principal Developer,
Software & Machine Learning,
FUJIFILM VisualSonics, Inc.

Abhishek Khowala
AI Solutions Architect,
Health and Life Sciences,
Intel Corporation

Table of Contents

Executive Summary

FUJIFILM VisualSonics (VSI) Artificial Intelligence-based measurement tools are extremely valuable for pre-clinical researchers analyzing small animal models in studies of human disease. Taking advantage of micro-ultrasound technology, these tools free researchers from time-consuming and error-prone manual procedures.

Like many AI tools, VSI's AutoLV Analysis software is highly compute-intensive and makes significant performance demands on underlying Intel Core processors to keep automated analysis user-friendly. Recently, VSI and Intel partnered to maximize the performance of AutoLV Analysis software by optimizing Deep Learning models using the Intel Distribution of OpenVINO Toolkit, resulting in increased ease of deployment and tremendous inference performance improvement.

41.4x
Greater Inferencing Performance[1]

FUJIFILM: Advancing Ultrasound Technology

VisualSonics designs and manufactures the world's highest resolution ultrasound and photoacoustic instruments. These systems operate at up to 70 MHz, enabling customers to image structures as small as 30 microns—features that are invisible to conventional ultrasound devices.

Used in many areas of pre-clinical research, VSI products enable researchers to study live animals in real-time, longitudinally, while eliminating safety issues and side effects encountered with other imaging modalities.

VSI is a subsidiary of FUJIFILM Sonosite, and both belong to a group of companies under FUJIFILM Healthcare. Sonosite also produces ultrasound products to serve different markets. VSI designs and develops tools for pre-clinical research, whereas Sonosite provides point-of-care ultrasound systems and medical informatics to physicians and clinicians, with the goal of enabling them to improve procedure efficiency, time-to-diagnosis, and patient outcomes.

Sonosite and VSI partner closely to enable technologies to migrate from pre-clinical research directly to important patient care solutions.

Left Ventricle (LV) Analysis

Cardiovascular researchers represent the largest base of VSI's pre-clinical customers. Left Ventricle (LV) analysis plays a crucial role in research aimed at alleviating human diseases. The metrics revealed by LV analysis enable researchers to understand how experimental procedures are affecting the animals they are studying. LV analysis provides critical information on one of the key functional cardiac parameters, *ejection fraction*, which measures how well the heart is pumping out blood and is key to diagnosing and staging heart failure. LV analysis also calculates several other standard cardiac function parameters, including *fractional shortening, stroke volume, and cardiac output*. A thorough understanding of these factors helps researchers to produce valid, valuable study results.

In addition, LV analysis has a growing role to play in the clinical care of human patients. The ability to display critical cardiac parameters in real time enables medical care providers to make a diagnosis more quickly and accurately during ultrasound interventions, without needing to stop and take measurements manually or to send images to the radiology department.

AutoLV Analysis Software

Efficient, reproducible analysis of imaging data is critical to research goals, both in the public and private sectors. **AutoLV Analysis Software**, VSI's Artificial Intelligence-based measurement product, is a fast and accurate tool for analysis of cardiovascular imaging data.

Building on VSI's widely adopted LV Analysis Tool, AutoLV Analysis brings AI to the functional analysis of the left ventricle in small laboratory animals with a "one-click" solution for both B-Mode and M-Mode research. Measurement and analysis of imaging data requires a significant investment of time and can sometimes be subject to inter-operator variability. Reliable, reproducible measurement data is the key to understanding model animal anatomy and physiology, and for completing studies, publishing work, and all other aspects of small animal pre-clinical research. AutoLV Analysis software makes functional and anatomical analysis of the left ventricle fast, highly reproducible, and free from manual error.

M-Mode and B-Mode

AutoLV Analysis provides two automated approaches to left ventricle analysis: M-Mode AutoLV and B-Mode AutoLV.

An **M-Mode** ultrasound—the M stands for "Motion"—examines a line of motion over time. When used in echocardiography, M-mode displays the movement of the myocardium, enabling accurate real-time measurements of the thickness of the heart wall, internal diameter, and heart rate. These measurements enable the calculation of key heart parameters, including ejection fraction.

In using the M-Mode method, researchers must calculate cardiac functional parameters acquired from the parasternal long-axis view. Typically, this involves making manual measurements of the thickness of the interventricular septum (IVS) or the right ventricle (RVID), the left ventricular interior diameter (LVID), the left ventricle posterior wall (LVPW—see Figure 1) at both systole (;s) and diastole (;d), and the heart rate.

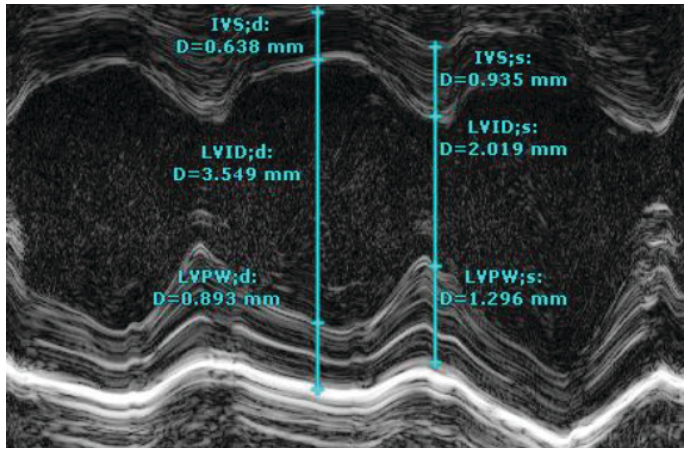


Figure 1. M-Mode thickness measurements of the anterior wall, chamber, and posterior wall—sometimes referred to as the "string method."

B-Mode (Brightness, or 2D mode) ultrasounds differ from M-Mode in that they show a single image at a given point in time: a two-dimensional ultrasound image composed of pixels representing ultrasound echo intensity. As in the case of M-Mode data, researchers can use B-Mode images to measure and quantify anatomical structure.

Both M-Mode and B-Mode are critical in assessing cardiac function, and both present problems for researchers. The challenge is that acquiring M-Mode or B-Mode measurements manually is laborious, time-consuming, and subject to human error, especially considering that both multiple systolic and diastolic points must be measured to provide data for cycle averaging.

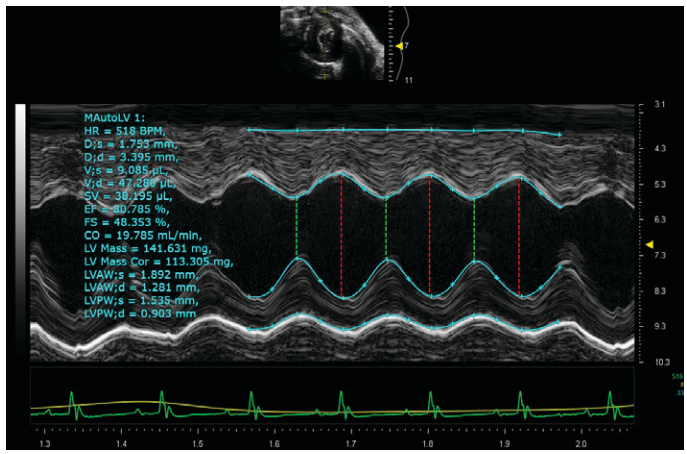
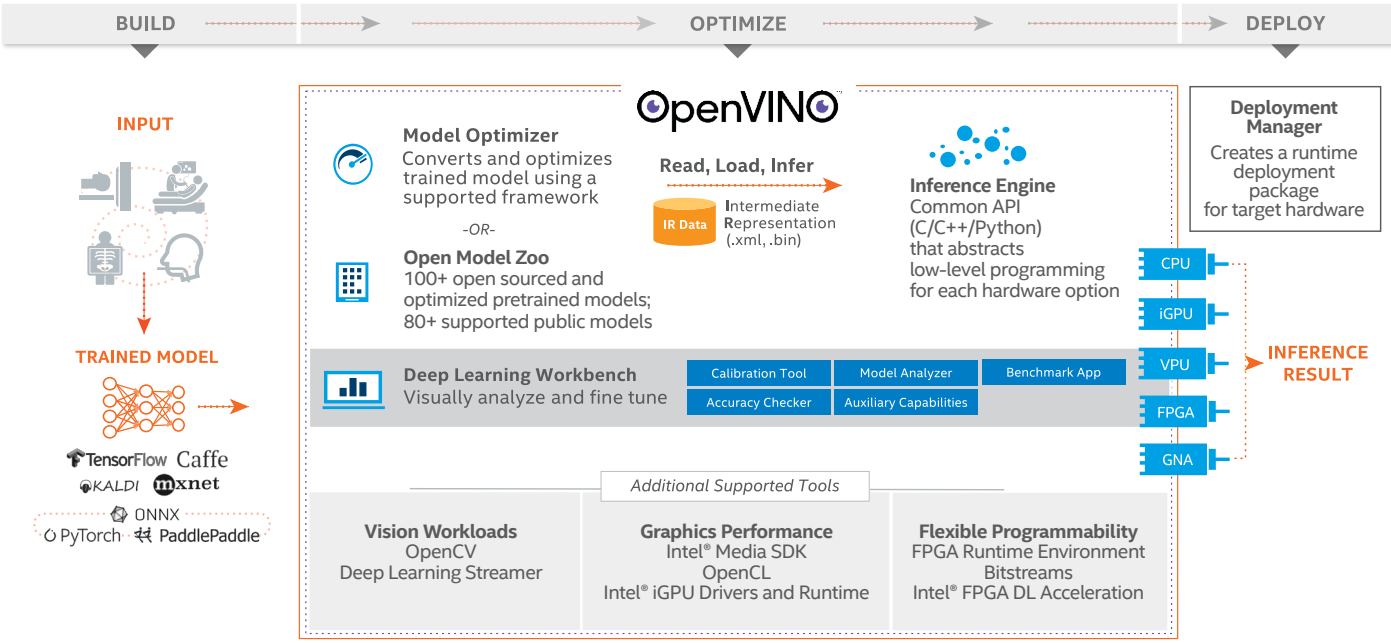


Figure 2. 4-Wall LV Trace of the endocardial border across multiple cycles.

Answering the Challenge

VSI invented AutoLV Software to help solve these problems for researchers. AutoLV automates the measurement process, removing the element of human subjectivity. These completely automatic measurements capture real-time details of the endocardial and epicardial borders (see Figure 2), facilitating the rapid calculation of cardiac functional parameters. Because it requires virtually no user intervention, AutoLV Software reduces both errors and the time required to achieve usable results.

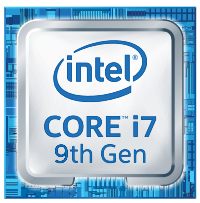
Under the Hood: Intel Distribution of OpenVINO Toolkit



The problems that AutoLV Software solves are challenging ones, due to both the inherent variability of ultrasound images and the difficulty of creating an algorithm that can accommodate this variability with sufficient accuracy to produce usable, valuable results. At this time, no other product on the market provides the features and functionalities that AutoLV Software delivers.

Intel Technology Makes It Possible

As an AI tool, AutoLV requires a platform that is powerful, flexible, and reliable. Both VSI and Sonosite rely on systems equipped with Intel® technology, including Intel Core i7 and i5 processors, as well as Intel Atom® processors. Running these systems enables VSI software developers to take advantage of the Intel Distribution of OpenVINO Toolkit. The toolkit, designed for Open Visual Inference and Neural Network Optimization (OpenVINO), makes it possible to harness the full potential of AI and computer vision. Based on Convolutional Neural Networks (CNN), the toolkit extends processing across Intel hardware (including accelerators) to maximize performance on demanding workloads, such as those AutoLV Software generates.



AutoLV would not be practical (or user-friendly) without the performance Intel technology delivers. Prior to the toolkit's availability, engineers at VSI could run deep learning solutions only as a post-processing operation with inference times on the order of hundreds of milliseconds per frame on the system CPU. With the introduction of the OpenVINO feature set, VSI can now optimize models directly for the CPU and integrated GPU. This has resulted in a **41.4x speedup** in inferencing processing.[1]

To help generate these improvements, VSI also took advantage of the Intel® Integrated Processing Primitives (Intel® IPP) library, an extensive set of ready-to-use, domain-specific functions that are highly optimized for diverse Intel architectures. Using Single Instruction Multiple Data (SIMD) instructions, the library helps improve the performance of compute-intensive workloads and accelerates processing speed while simplifying code development.

In addition, they leveraged OpenCV, a complimentary toolkit optimized for Intel architecture. Working in conjunction with Intel IPP, OpenCV improves the processing of real-time images and provides additional analytics and deep learning capabilities.

The implementation of these Intel technologies, both on the hardware and the software side, makes it possible to offer imaging applications in real time, even when using only an Intel CPU without relying on the integrated GPU.

> The purpose of FUJIFILM VisualSonics: "Through bold innovation, we empower those dedicated to the advancement of human health."

FUJIFILM
VISUALSONICS

The VisualSonics Solution

To establish the viability of AutoLV as a practical application, tests were performed on hardware similar to that found in actual laboratory environments. Intel Core i7 processor-based systems were used for the tests. The tests ran Deep Learning methodologies (Artificial Intelligence and Neural

Networks) trained on research images generated at VSI.

The M-Mode AutoLV algorithm was then used to compare performance. Deep Learning inference optimization was compared using models converted for 1) TensorFlow, and 2) the Intel® Distribution of OpenVINO™ toolkit.

The Solution Model

The key aspect of AutoLV is the automatic measurement of the interior (endocardial) and exterior (epicardial) heart wall boundaries in M-Mode and the interior wall boundary in B-Mode. These boundaries can be used to obtain the measurements needed to calculate cardiac functional metrics. Because of the extreme variations in input data and the difficulty in trying to describe a procedure for performing these traces, deep learning was leveraged to provide a solution. Alternatives to deep learning were considered, including non-AI-based segmentation algorithms, but past experiences have shown that those lack the needed accuracy and consistency. Neural networks, on the other hand, have empowered thousands of applications where other methods have failed, and therefore were selected for the application.

For a deep learning algorithm to succeed, three features are required: an abundance of labeled input data, a suitable deep learning model, and successful training of the model parameters.

M-Mode Solution Process

Data for M-Mode training consisted of VisualSonics data sets from a variety of animal models acquired under different conditions. Collected over a period of many years, this data represents a very diverse set of input conditions. Approximately 2,000 data sets were selected, and Vevo LAB was used to carefully label the four walls in each segment for use in training, as shown in Figure 3. 4-8 heart cycles were traced for each sample, with each data set taking approximately 1-2 minutes to trace manually.

To generate training data, a completely traced M-Mode region was divided into smaller "chunks" of data and separated into wall-pairs. Each chunk was treated as an individual training example. A complete region could be

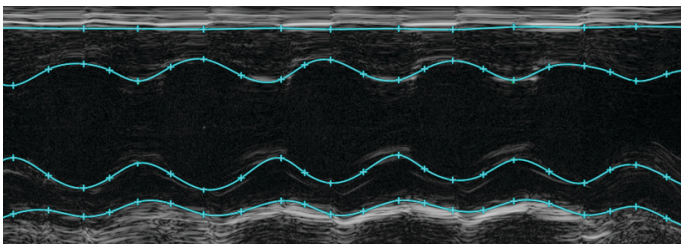


Figure 3. Four walls labeled.

several heart cycles and perhaps 500ms. long; breaking it into chunks, each piece might be as little as 100ms. The training set thus consisted of approximately 10,000 unique training examples from 2,000 unique data sets. Data was formatted using C++ and Python for use in an end-to-end TensorFlow/Keras training framework.

Models were set up to handle inner walls separately from outer walls; while these could have been designed to detect all four walls at the same time, the decision was made to handle them separately to increase system flexibility.

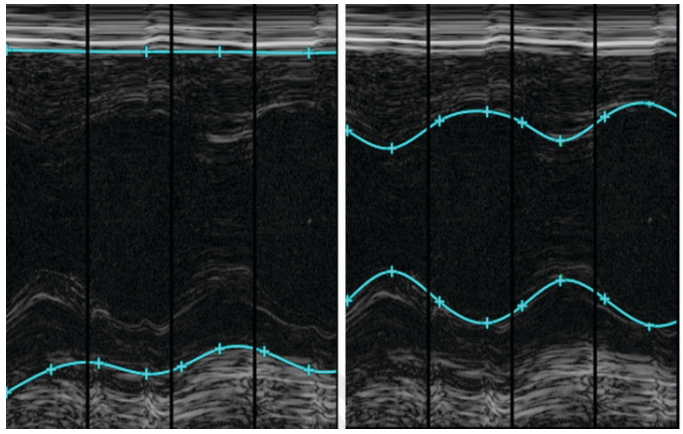


Figure 4. Four "chunks" of outer wall and inner wall pairs.

The final processing flow is shown in Figure 5. Initially, a MobileNet V2-based model was developed[2], which took a raw M-Mode chunk of data formatted as a 256 x 128 image as input and then output the relative top and wall positions directly for each vertical line in the image (256 * 2). (See Figure 6a and 6b.)

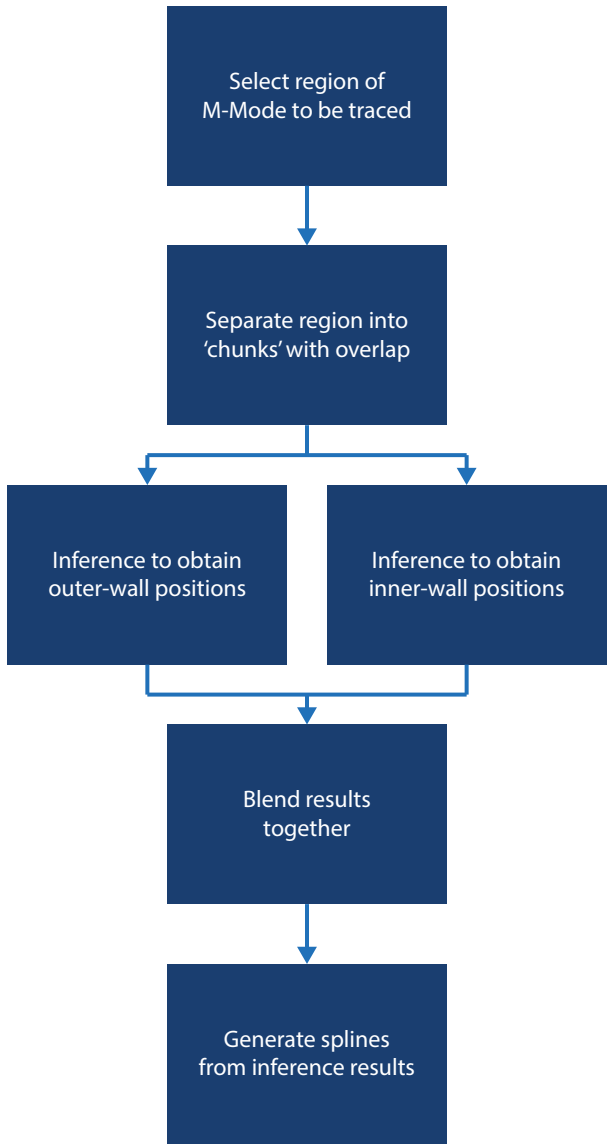


Figure 5. M-Mode processing methodology using MobileNet.

TensorFlow vs. Intel Distribution of OpenVINO Toolkit

After the model was developed, a compiled version of TensorFlow running on a Windows system was used to perform the inferencing, which ran only on the CPU. The result was extremely complex and time consuming. Compiling TensorFlow from source requires a number of external tools and generates a large, cumbersome set of libraries. In addition, inference times are slow. Tracing an entire data set requires a number of chunks (10 or more) to be processed, and the processing time for an entire data set could be several seconds.

Next, the Intel Distribution of OpenVINO Toolkit was used for the inference workload. The toolkit's library was incorporated into the test application, and Model Optimizer was used to convert the original trained model to Internal Representation (IR) format. The results were very different than those seen with TensorFlow; they were significantly simpler, and inference times were reduced to 3.1 ms.[1] In addition, the toolkit enabled us to run half the inference operations on the CPU and the other half on the processor's integrated GPU, dividing the processing between two parallel CPUs. What is especially significant here is that the integrated GPU was a low-power, general purpose GPU, similar to what is found on most consumer PCs—not a high end, high power GPU that might be found only on a more expensive AI system.

Input	Operator	t	c	n	s
256x128x1	conv2d	-	32	1	2
128x64x32	bottleneck	1	16	1	1
128x64x16	bottleneck	6	24	2	2
64x32x24	bottleneck	6	32	3	2
32x16x32	bottleneck	6	64	4	2
16x8x64	bottleneck	6	96	3	1
16x8x96	bottleneck	6	160	3	2
8x4x160	bottleneck	6	320	1	1
8x4x320	conv2d 1x1	-	1280	1	1
8x4x1280	glbavgpool 8x4	-	-	1	-
1x1x1280	dense 512	-	-	-	-
512	output	-	-	-	-

Figure 6a. MobileNet model architecture.

Improved inference times using the toolkit provided the freedom to develop more complex models without fear of overtaxing the hardware. An improved implementation of the M-Mode AutoLV system used a larger, more complex U-Net model to generate a segmentation region between the wall boundaries, as shown in Figure 7. The U-Net was able to generate segmentation regions that were more accurate than the original MobileNet solution.

The U-Net model was trained to segment the region within the wall boundaries. The inference stages were the same as with the original MobileNet model. However, instead of getting the relative wall positions directly from the model, a segmentation of this region was used. The modified methodology is shown in Figure 8. It was not difficult to trace the top and bottom edges of the segmentation map to obtain the final wall positions. To further optimize inference times, the size of the chunks was increased to require fewer inferences.

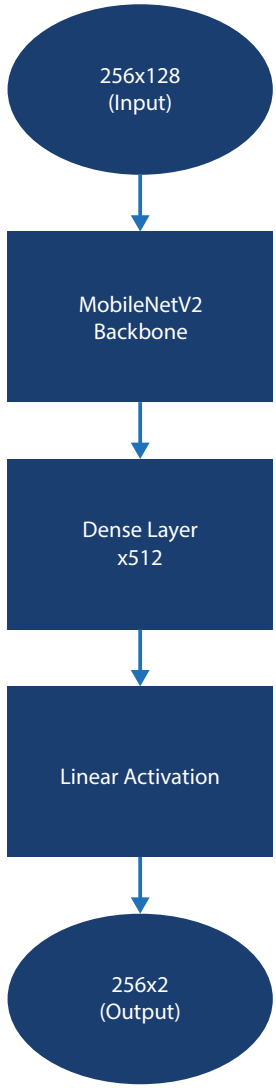


Figure 6b. MobileNet model architecture.

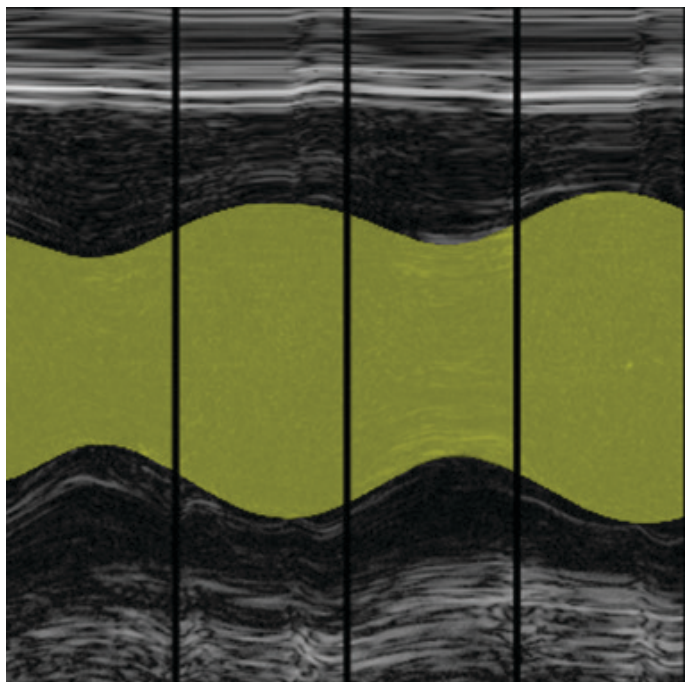


Figure 7. Segmentation of inner walls.

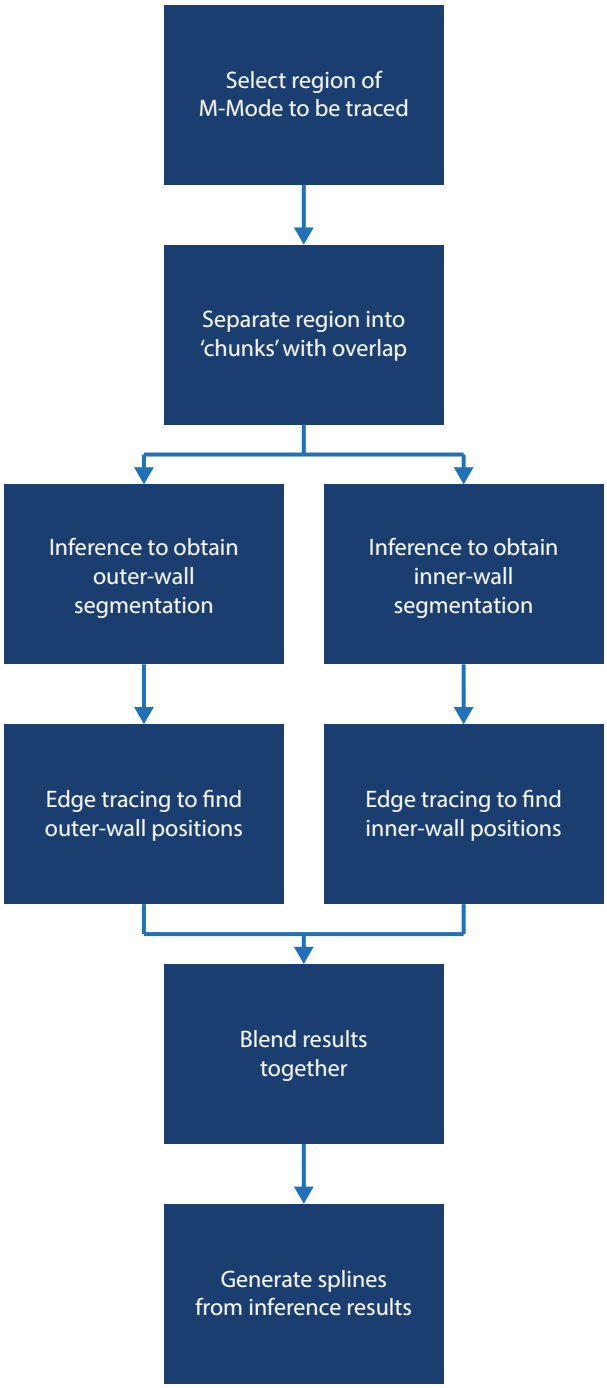


Figure 8. Updated M-Mode processing methodology using U-Net.

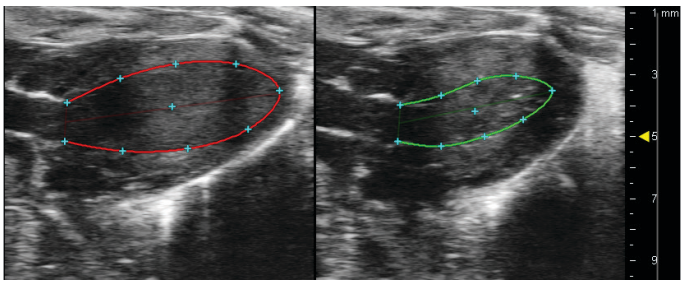


Figure 9. Tracing boundary walls in B-Mode.

B-Mode Solution Process

Tracing wall boundaries in B-Mode is a difficult task. In this case, all frames had to be traced between systole and diastole (see Figure 9).

Approximately 2,000 PSLAX data sets were collected in the parasternal long-axis view, and the inner wall boundaries were traced over a number of cycles. The acquisition frame rate, which depended on the transducer and imaging settings used, varied from 100 to 1,000 fps. That meant 30 to 100 individual frames were traced for each cine loop. By the end, a collection of over 150,000 unique images were assembled for training. Training augmentation included horizontal flip, noise, contrast, brightness, and deformable image warp.

In the interest of efficiency and simplicity, a 2D model design was chosen to trace each frame individually. While a 3D model could be used (or perhaps even a recurrent model to handle the video nature of the moving heart), a 2D model was used because it enabled an accurate but much simpler training system. A U-Net model with an input and output size of 128 x 128 was trained on a segmentation map of the inner wall region. The total number of parameters for this model was 2,736,451.

Training was performed on an NVIDIA VT100 GPU and took several hours to complete using the TensorFlow/Keras-based training framework.

The processing methodology is somewhat different than in M-Mode. While the inference is processed separately for each frame, what the user requires is a cine-loop of processed frames for a complete heart cycle. To accomplish this, after 2D inferencing of each frame in a cycle, post processing is performed to improve the overall accuracy (see Figure 10).

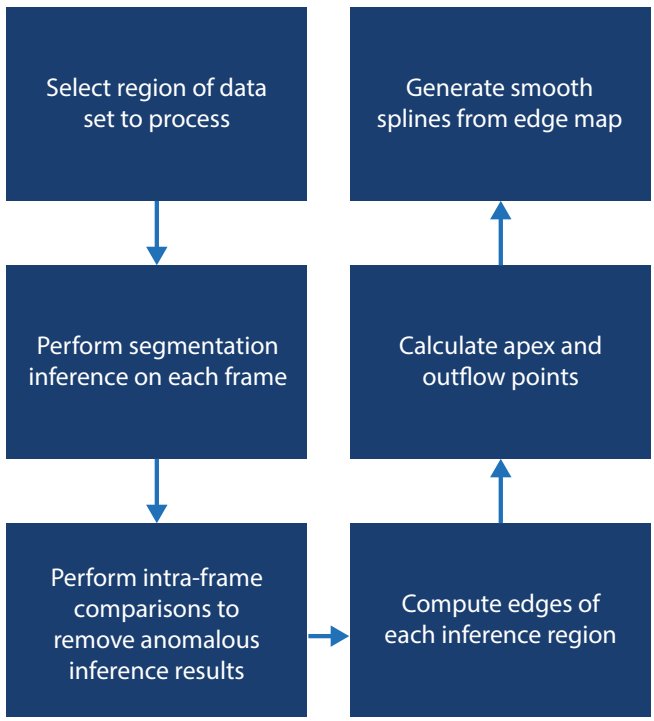


Figure 10. B-Mode processing methodology.

Performance Comparison

To evaluate the speedup enabled by the Intel Distribution of OpenVINO Toolkit, a test comparison was performed using the M-Mode MobileNet solution.

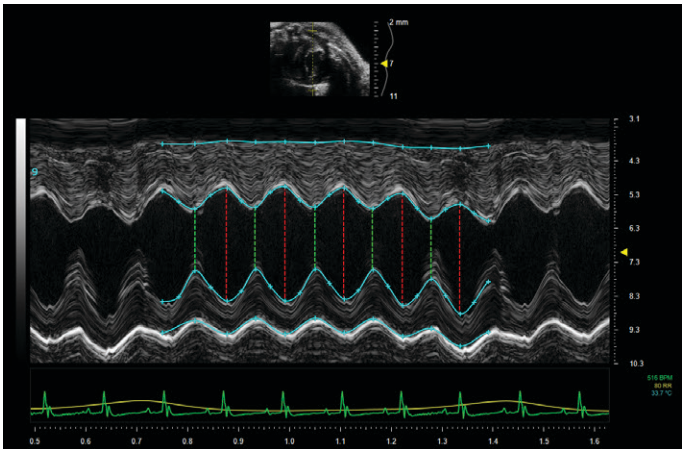


Figure 11. Image used to generate performance comparison. The trace duration was approximately 643ms.

The model used in testing was a close variant of MobileNet V2 used for the M-Mode solution. No model optimization was performed for the TensorFlow version. The IR version had default model optimizations applied.

TensorFlow version 1.4.0 was used, compiled with Visual Studio 2015, using full release optimizations (/O2 and /Ob2) and the Extended instruction set (/arch:AVX) for x64 bit. The final library size was 1.53 GB, including release and debug DLLs; it required 4,415 include files. The optimized Intel version of TensorFlow was not available for Windows and as such was not available for testing.

The **Intel Distribution of OpenVINO Toolkit** model used version 2019.2.242.

Tests were performed using Microsoft Windows 10, with no other applications running.

Comparison Results

Using Version 1 of the M-Mode AutoLV algorithm to compare the model converted for TensorFlow to the model converted using the Intel Distribution of OpenVINO Toolkit produced the results shown here.

TensorFlow Model

The TensorFlow image required running three inference blocks each for the inner and outer walls, for a total of 6 blocks. The test was run four times. Two models for the inner and outer sections were run in parallel, resulting in an overall performance improvement.

	Inner Wall	Outer Wall	Inner + Outer Parallel (2 Threads)
Trial 1 (MS)	773.7	776.0	776.2
Trial 2 (MS)	768.1	780.2	781.5
Trial 3 (MS)	765.2	777.3	777.5
Trial 4 (MS)	762.3	774.3	774.4
Average			777.8

The average time per inference (6 per trial) was **129.6 ms**.[1]

Intel Distribution of OpenVINO Toolkit Model

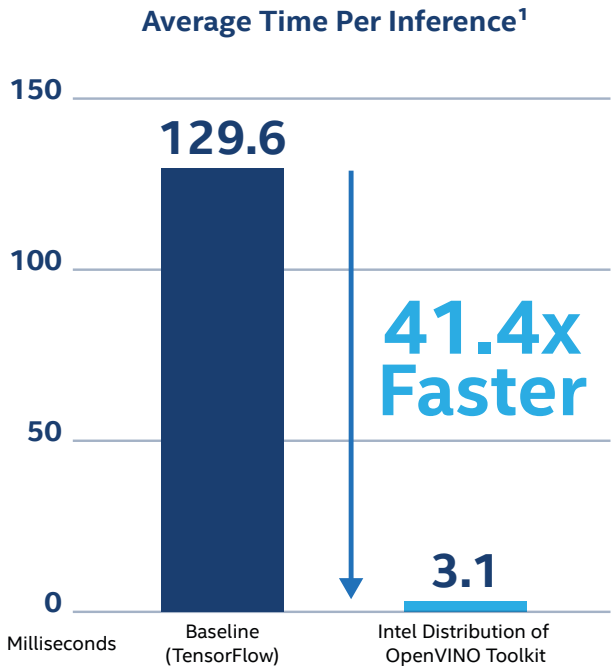
Because OpenVINO is very fast on tasks such as these, we decided to add more processing blocks, with extra overlap, in hopes of improving results. For the test image, 16 inference operations were required for each of the inner and outer walls (32 operations in total). Processing was done as a review (not as a real-time operation). No attempts were made to split processing between the CPU and the integrated GPU, opting only for CPU inference. In addition, the inner and outer walls were processed synchronously (though each of those inference operations was threaded internally by the toolkit's library).

	Inner Wall	Outer Wall	Inner + Outer
Trial 1 (MS)	45.2	45.9	91.2
Trial 2 (MS)	46.7	45.0	91.8
Trial 3 (MS)	44.8	47.9	92.9
Trial 4 (MS)	59.8	64.6	124.6
Average			100.1

The average time per inference (32 per trial) was **3.1 ms**.[1]

Conclusion

As these charts make clear, the TensorFlow model test resulted in an average time per run of 129.6 ms. The Intel Distribution of OpenVINO Toolkit test, on the other hand, resulted in an average time per run of 3.1 ms. In other terms, the toolkit's model, running on an Intel Core i7 processor-based system, ran **41.4x times faster** than the TensorFlow model.[1]



Working Together for a Better Future

Two FUJIFILM companies, VisualSonics and Sonosite, with very different markets and customers, have found common ground upon which to build tools that will lead to improvements in human health.

FUJIFILM Sonosite develops clinical ultrasound systems with particular emphasis on point-of-care and portable systems. Sonosite products are used by physicians and other medical staff in the treatment of patients. These clinicians are continually looking to deliver improved patient care resulting in better patient outcomes.

These two companies maintain constant communications and collaborate closely about the ever-changing technology they work with. Improvements discovered by one group often have real and immediate value for the other group. It's certain that the success of the Intel Distribution of the OpenVINO Toolkit model running on Intel® processor-based hardware will prove to be as relevant for the clinical side as it has been for researchers.

In addition, Intel Corporation is constantly developing new ways to harness silicon designed specifically for AI, both in stand-alone applications like AutoLV Software and in end-to-end solutions that span from the data center to the edge. As the tests outlined above make clear, Intel architecture-based systems are delivering on the promise of AI today, with much more to come.

1. Performance results are based on testing as of May 2020 by FujiFilm, and may not reflect all publicly available security updates. No product can be absolutely secure. Test system configuration: Intel Core i7 processor 6700HQ, 2.6GHz , Dell Model 0XC72F-A00 Motherboard, 16 GB Dual Channel memory, integrated Intel HD Graphics 530, Microsoft Windows 10 Build 14393.
2. https://arxiv.org/abs/1704.04861, https://arxiv.org/abs/1801.04381

FUJIFILM VisualSonics:
Visualsonics.com/about-us/our-company

FUJIFILM Sonosite:
Sonosite.com/about

Intel Core Processors:
Intel.com/core

Intel AI Technologies:
Intel.com/ai

Intel Distribution of OpenVINO Toolkit, Powered by OneAPI:
Intel.com/openvino

Intel Health and Life Sciences Technologies:
Intel.com/healthcare