Deep Learning for Lung Cancer Detection

Intel® Distribution of OpenVINO™ Toolkit Speeds Up Lung Nodule Detection and Segmentation on Intel® Xeon® Scalable Processor-based Platforms

Authors

Predible Health

Kiran Vaidhya

Adarsh Raj

Krishna Chaitanya

Abhijith Chunduru

Suthirth Vaidya

Intel Corporation

Dr. Ramanathan Sethuraman

Madhu Kumar

Dmitry Rizshkov

Prashant Shah

Introduction

With an annual incidence of 3.34 million, lung cancer is the deadliest cancer, with an estimated 1.88 million deaths per year worldwide[1]. Early detection is critical towards long-term survival; stage 4 lung cancer has a 5-year survival rate of 5% in comparison with stage 1 lung cancer that has a 5-year survival rate of 56%[2]. The National Lung Screening Trial (NLST) revealed that participants who received low-dose helical CT (computed tomography) scans had a 20 percent lower risk of dying from lung cancer than participants who received standard chest X-rays[3]. Advances in multi-detector CT scanning have made high-resolution volumetric imaging possible in a single breath hold, at acceptable levels of radiation exposure[4]. Several observational studies have shown that a low-dose helical CT scan of the lung detects more nodules and lung cancers, including early-stage cancers[5]. Potentially malignant lung nodules can be identified from chest CT scans, and early intervention can result in a higher chance of long-term survival[6].

A typical chest CT scan contains anywhere in the range of 300-500 slices, and a radiologist must examine each slice to detect lung nodules. Lung nodules are small masses of tissue in the lung that appear as round, white shadows on a CT scan and are often difficult to detect and document. Most are benign[7], but their detection requires specialized expertise and with widespread implementation of lung cancer screening programs, the burden on radiologists is rapidly increasing.

Computer-aided-detection (CAD) is becoming increasingly useful in helping radiologists interpret high-dimensional imaging data like CT and MRI scans. CAD algorithms have also shown to be successful in increasing radiologists' ability to detect lung nodules[8][9]. With the advent of deep learning and convolutional neural networks (CNNs), CAD algorithms have started moving away from a reliance on hand-crafted features requiring custom engineering, to learning features from data through CNNs[10][11].

> "Predible Health's deep learning solutions have consistently demonstrated improved precision and efficiency for Radiologists, especially in cancer screening settings. Our collaboration with Intel enables us to deploy within the hospital premises, ensuring seamless workflow integration and real-time inference of the studies."
>
> – Suthirth Vaidya, CEO, Predible Health

This whitepaper will detail how Predible Health's deep learning algorithm for detecting lung nodules on CT scans has been optimized on Intel® Xeon® Scalable processors using the Intel® Distribution of OpenVINO™ Toolkit.

Algorithm, Workflow, Requirements

Predible has built a software tool that automatically queries chest CT DICOM images from a picture archiving and communication system (PACS), processes them, and uses neural networks to detect lung nodules. DICOM (Digital Imaging and Communications in Medicine) is an industry-standard method for handling, storing, printing, and transmitting medical imaging information. Once the DICOM series has been processed by the neural networks, the results are sent back to the PACS and are available to be viewed by the radiologist.

The deep learning system was trained to detect and segment lung nodules from chest CT scans. The system contains three stages, and each stage uses a combination of neural networks to solve a specific subtask.
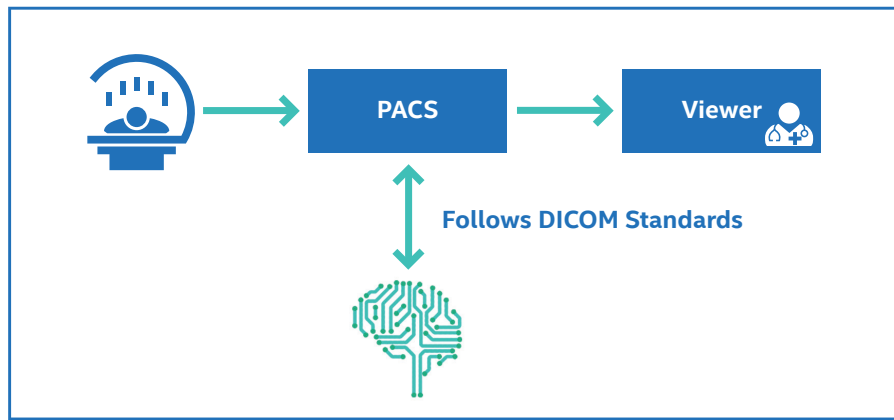
Table of Contents

Figure 1. Automatic Query & Retrieval (AutoQR) pipeline integrates with PACS for processing Chest CT DICOM images.
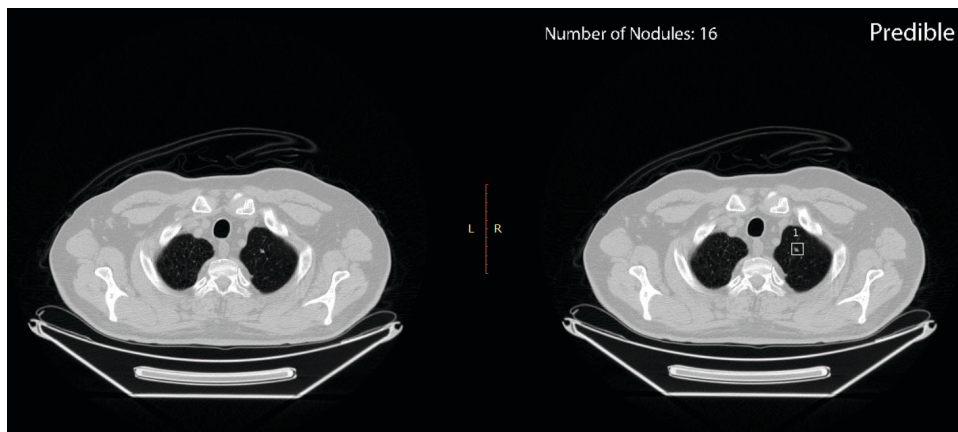

Figure 2. A lung nodule detected by the deep learning system. L: Original image; R: Annotated image. Images used with permission by Predible Health.
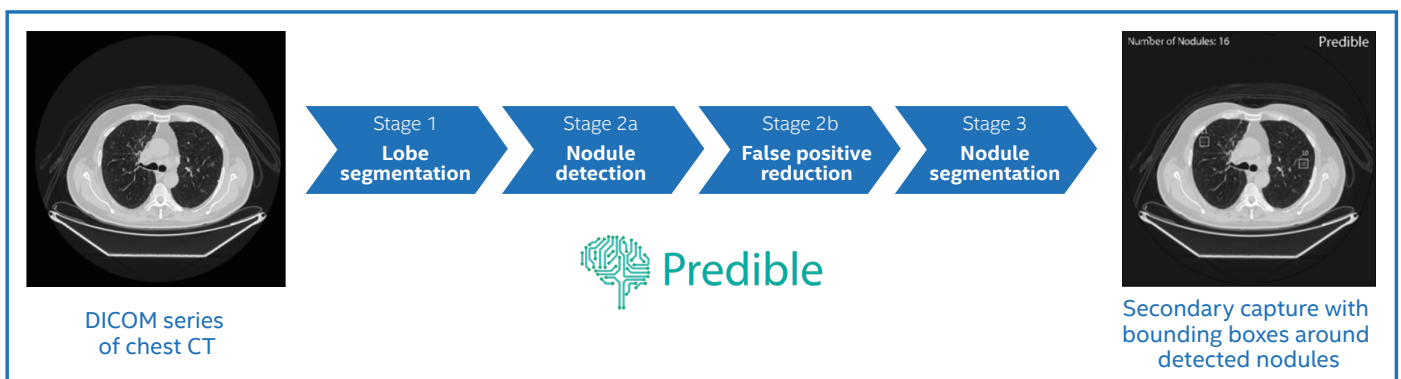

Figure 3. Architecture of the neural network pipeline for detecting lung nodules from chest CT. Images used with permission by Predible Health.

Stage 1: Stage 1 comprises a 3D U-Net (a convolutional neural network that was developed for biomedical image segmentation)[12] which performs semantic segmentation of five different pulmonary lobes from the incoming chest CT. Once the chest CT is retrieved from the PACS, the CT scan is sent through Stage 1 where the lobe segmentation maps are used to extract the lung region from the chest CT.

Stage 2a: Stage 2a includes three 3D Faster R-CNNs. While convolutional neural networks (CNNs) are used for image classification, R-CNNs (the R stands for region) are used for object detection. In this case, R-CNNs are assigned to detect nodules in 3D with a high sensitivity. We chose to focus on detecting nodules between 3mm and 30mm in size. Nodules less than 3mm in size are irrelevant according to lung cancer

screening protocols, and nodules greater than 30mm in size are large enough to be detected by a naked eye[13]. Minimal pre-processing is done after cropping the lung region using the lobe segmentation maps. The pre-processed lung image is sent through Stage 2a, where the ensemble scans through the 3D volume to detect lung nodules varying from size 3 to 30mm. The outputs from each network in the ensemble are combined through non-maximum suppression to provide a final list of candidate nodules with a high sensitivity.

Stage 2b: Stage 2b consists of an ensemble of 3D wide residual networks[14] whose task is to classify each nodule proposal from Stage 2a as a nodule or false positive.

Stage 3: Stage 3 is comprised of a 3D U-Net whose task is to segment and compute volumes of the detected nodules from Stage 2. The detected nodules from Stage 2 are processed one-by-one with the nodule segmentation network. The

segmentation maps are then utilized to estimate volumetry.

The centroid and diameter for each candidate is then used to draw rectangular bounding boxes in each slice of the detected nodules. The final image is compiled together as a DICOM series and then is pushed back to the same study in the PACS using the AutoQR system.

The secondary capture resides in the same study as an additional series that can be opened by the radiologist. The bounding boxes drawn around detected lung nodules can serve as an additional pair of eyes for the radiologists and can help them in detecting subtle lung nodules that they are likely to miss. Additionally, a DICOM report is added to the secondary capture to summarize the detections and to quantify the presence of emphysema using the lobe segmentation maps.
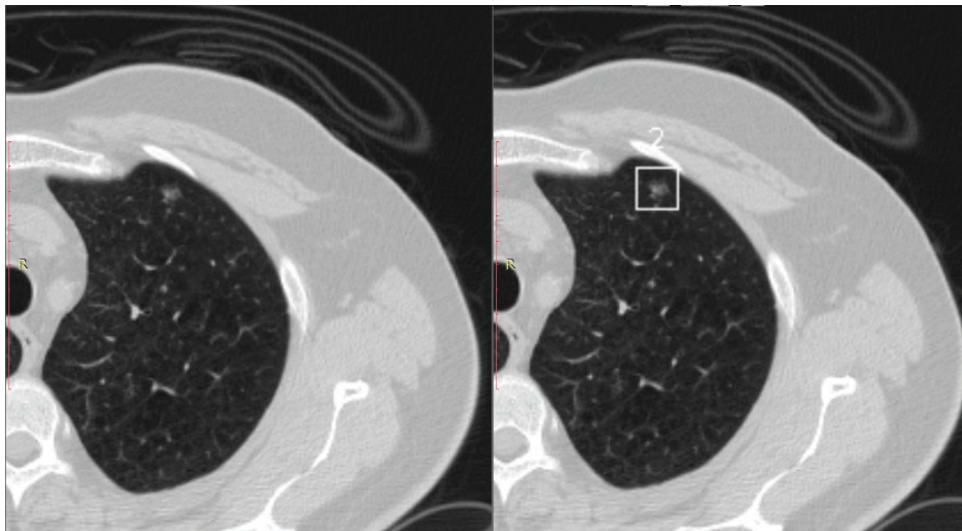

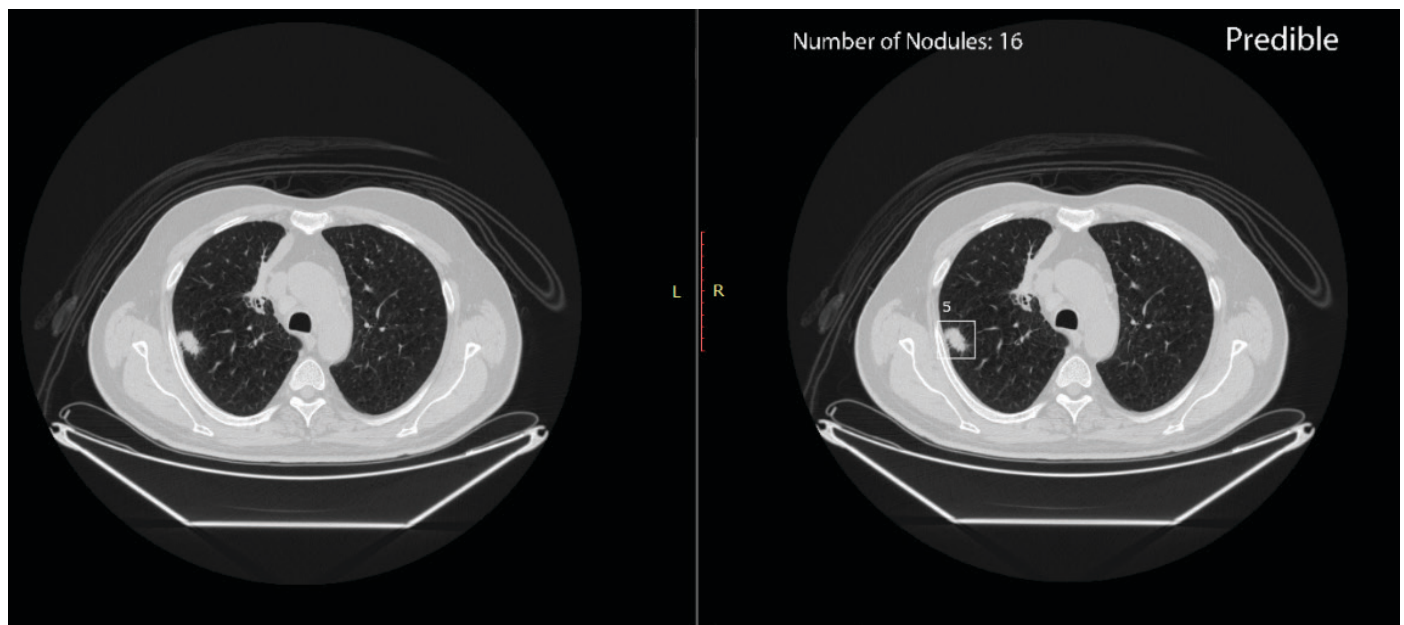**Figure 4.** Subtle part-solid nodule detected by the algorithm. Images used with permission by Predible Health.

Figure 5. Highly suspicious lesion detected in the right lower lobe. Images used with permission by Predible Health.
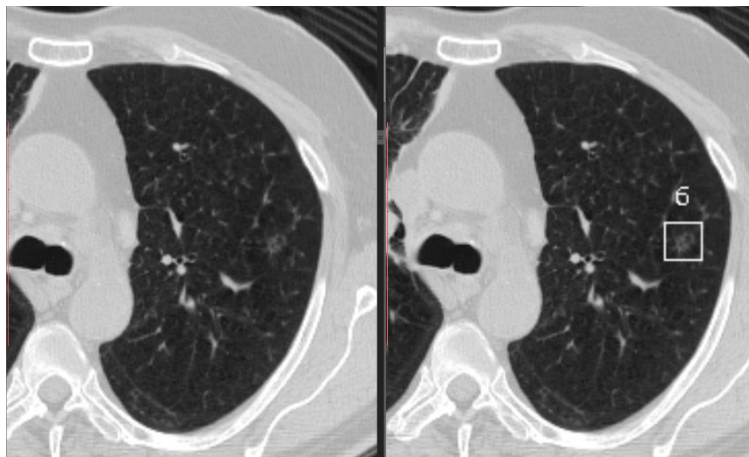
Figure 6. Subtle ground-glass nodule detected by the algorithm in the left lung. Images used with permission by Predible Health.
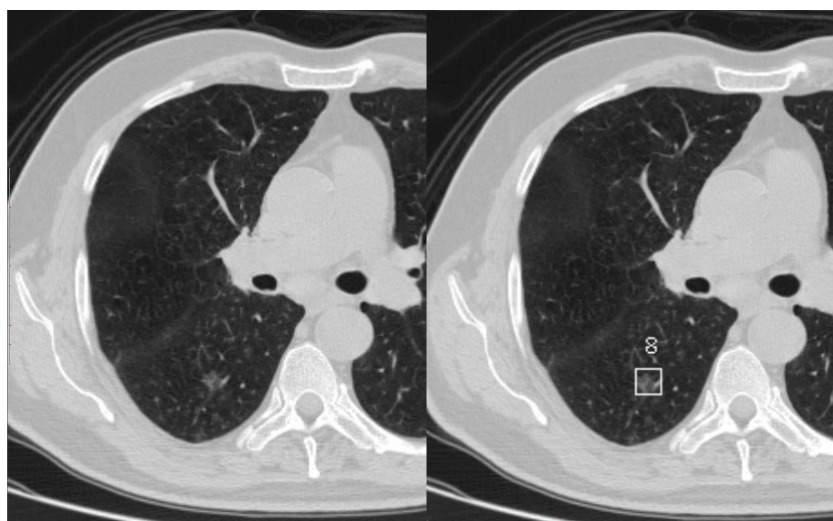
Figure 7. Part-solid nodule detected in the right lung. Images used with permission by Predible Health.

Predible

Number of Nodules: 16

ID	Slice	Volume	Slice
2	43	0.376mL	9mm
5	61	5.31mL	24mm
6	75	0.527mL	10mm
8	93	0.575mL	5mm
14	145	0.348mL	8mm

Emphysema Quantification %LAV950

Whole lung	5.25%	Right lower lobe	1.7%
Left lower lobe	2.21%	Right middle lobe	7.03%
Left upper lobe	6.34%	Right upper lobe	11.07%

Figure 8. Summary DICOM of lung nodule analysis and emphysema quantification. Image used with permission by Predible Health.

Analysis of Computational Requirements

Estimation of turnaround times (TATs) for radiology reporting is highly subjective and depends on the clinical setting[15]. Interpreting a chest CT scan for detection of cancer is not a time-critical requirement w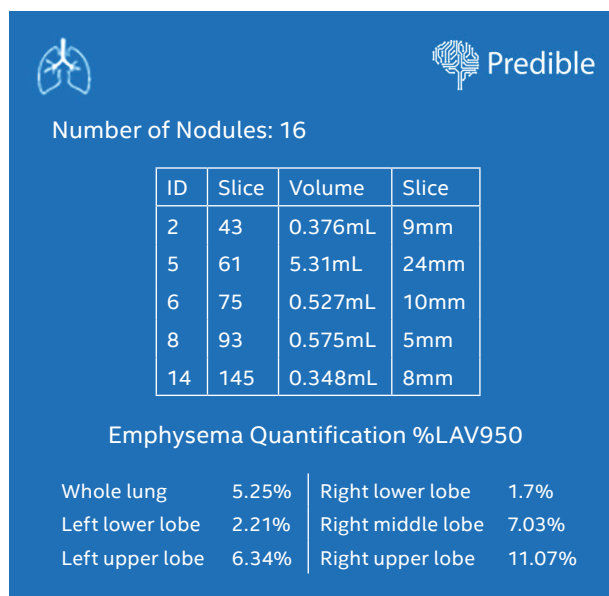hen compared to the likes of acute stroke protocol CT scans. However, for estimating computational requirements, we assume an operating window of one hour between start of image acquisition and start of interpretation by the radiologist. This means that the deep learning algorithms have about 30 minutes to process a chest CT scan and push the resulting secondary capture onto the PACS, which leaves 30 minutes for image acquisition. In hospitals, we expect use of either dedicated or shared compute assets for deep learning-based inferencing. In the former case, the expense of a dedicated compute asset could limit options for performance, resulting in a longer time to infer, while in the latter case, compute resource sharing could result in longer or shorter inference times, based on the clinical workflow.

For this study, the patches optimized by the Intel Distribution of OpenVINO toolkit for image recognition in workflow stages 1-3 are executed on Intel Xeon Scalable processors and the performance is compared with a non-optimized PyTorch* software baseline on the same compute platform. Other pre-processing and post-processing stages are not compared since they are typically executed on Intel® processor-based systems that already deploy optimized libraries. Depending on the compute asset choice (dedicated versus shared), the pre- and post-processing stages can take several minutes. This leaves several minutes for the various deep learning stages to process.

Analysis on 1010 CT scans from the Lung Image Database Consortium and Image Database Resource Initiative (LIDC-IDRI)[13] showed that an average chest CT consists of 240 slices (median=207) with a median slice thickness of 2mm. For estimating computational requirements, it is advisable for profiling to be done on the largest image possible. Most commonly occurring chest CT scans have a slice thickness of around 2mm and a total of around 200 slices. Some large images have a slice thickness of 1mm with a total of around 600 slices. For this study, we assumed that the largest chest CT scan consists of 700 slices with a slice thickness of 1mm. We also assumed that this CT scan had a total of 30 nodules detected by our algorithms for the same reason.

Intel® Distribution of OpenVINO™ Toolkit Optimization

Enabled by tools like the Intel Distribution of OpenVINO toolkit, Intel Xeon Scalable processors offer a flexible platform for AI model inferencing. The toolkit's offline Model Optimizer (MO) optimizes graph-level constructs 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. In the online process, the toolkit's Inference Engine optimizes MO output based on the target hardware: Intel® Xeon® processors, Intel® Core™ processors, Intel® Processor Graphics, an Intel® field-programmable gate array (FPGA), or Intel® Movidius™ Myriad™ vision processing units (VPUs). The Intel® Math Kernel Library for Deep Neural Networks (Intel® MKL-DNN) is an open-source performance library that significantly boosts performance of deep neural networks on Intel® CPUs.
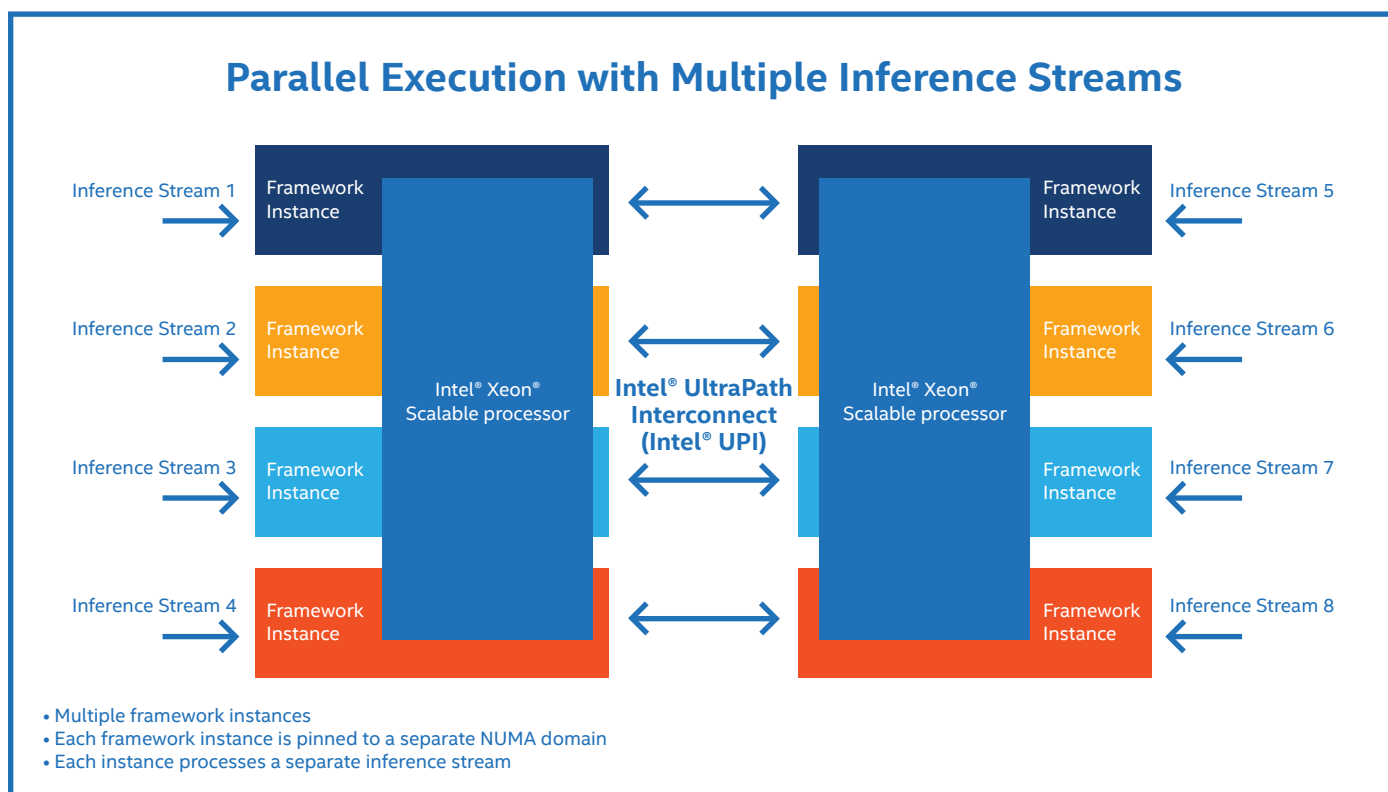

Parallel Execution with Multiple Inference Streams

Inference Stream 1 → Framework Instance

Inference Stream 2 → Framework Instance

Inference Stream 3 → Framework Instance

Inference Stream 4 → Framework Instance

Intel® Xeon® Scalable processor

Intel® UltraPath Interconnect (Intel® UPI)

Intel® Xeon® Scalable processor

Framework Instance ← Inference Stream 5

Framework Instance ← Inference Stream 6

Framework Instance ← Inference Stream 7

Framework Instance ← Inference Stream 8

- Multiple framework instances
- Each framework instance is pinned to a separate NUMA domain
- Each instance processes a separate inference stream

Figure 9. Sub-socket partitioning across dual-socket Intel® Xeon® platforms for multiple inference streams.

Further performance gains can be obtained by running multiple instances of the toolkit on each of the sockets of a CPU (see Figure 9), instead of running just one instance of the toolkit in both sockets. Each instance is bound to one or more cores, which results in better core utilization. For example, as depicted above, eight instances of the toolkit are run on two Intel Xeon Scalable processor sockets. In the Predible model, each patch (3D volume) per stage can be an input to an instance of the module, thereby exploiting multiple instances of the module running in parallel.

Performance Comparison

Algorithmic Performance

The lobe segmentation models were trained on thoracic CT scans obtained from the Lung Tissue Research Consortium (LTRC) dataset[16]. The model showed an average dice coefficient of 0.95 for segmenting each lobe on 170 CT scans.

The nodule detection ensemble was trained on multi-institutional data obtained from multiple hospitals in India and the US amounting to a total of 1,500 densely annotated CT scans from a cohort of 80,000 CT scans. The ensemble was evaluated on an unseen set of 177 CT scans from the LIDC-IDRI dataset and showed a performance of 89% sensitivity in detection of 3mm-30mm nodules at a specificity rate of one false positive per CT scan when the predicted centroids were compared against a ground-truth consensus of three radiologists.

The nodule segmentation model was trained and validated on the LIDC-IDRI dataset. The model showed an average dice coefficient of 0.68 when compared against the intersection of contours annotated by the radiologists participating in the LIDC-IDRI[13] study.

Compute Performance

We chose three different Intel Xeon Scalable processor SKUs as target implementation hardware. The Peak TFLOPS (single precision – FP32) for each processor are listed in Table 1. All inferencing models use FP32 processing. Complete hardware and software configurations used for testing are provided in Appendix A.

Processor	Peak TFLOPS (FP32)
Intel® Xeon® Silver 4110 Processor	1.1
Intel® Xeon® Gold 6140 Processor	5.3
Intel® Xeon® Platinum 8168 Processor	8.3

Table 1. Peak TFLOPS (FP32) for Intel® Xeon® processors.

For the defined largest image, Figure 10 shows performance gains achieved through optimizations compared to the baseline implementation. As can be seen, the various modules in different stages of the Predible Lung CT model showed significant performance improvements. Three major optimization steps contributed to the performance improvements:

1. Intel Distribution of OpenVINO toolkit-based inferencing model optimization

2. Multi-instance Intel Distribution of OpenVINO toolkit running on multiple CPU sockets

3. Custom inferencing model optimizations

Custom inferencing model optimizations provided opportunities to merge multiple layers or expose more channels for processing that can then benefit from better hardware utilization. These custom optimizations are enabled in subsequent releases to benefit the broader AI model community.

Stage 1

Stage 2a.1

Stage 2a.2

Nodule Detection -2 (patches per second) — Baseline vs Optimized across Intel Xeon Silver CPU (15X), Intel Xeon Gold CPU (63X), Intel Xeon Platinum CPU (77X).

Stage 2a.3

Nodule Detection -3 (patches per second) — Baseline vs Optimized across Intel Xeon Silver CPU (13X), Intel Xeon Gold CPU (61X), Intel Xeon Platinum CPU (83X).

Stage 2b.1

False Positive Reduction -1 (patches per second) — Baseline vs Optimized across Intel Xeon Silver CPU (10X), Intel Xeon Gold CPU (44X), Intel Xeon Platinum CPU (67X).

Stage 2b.2

False Positive Reduction -2 (patches per second) — Baseline vs Optimized across Intel Xeon Silver CPU (10X), Intel Xeon Gold CPU (45X), Intel Xeon Platinum CPU (68X).

Stage 3

Nodule Segmentation (patches per second) — Baseline vs Optimized across Intel Xeon Silver CPU (17X), Intel Xeon Gold CPU (53X), Intel Xeon Platinum CPU (58X).

Figure 10. Performance of Intel Distribution of OpenVINO™ toolkit optimization versus PyTorch* baseline for various stages of Predible model on Intel® Xeon® Scalable Processors.
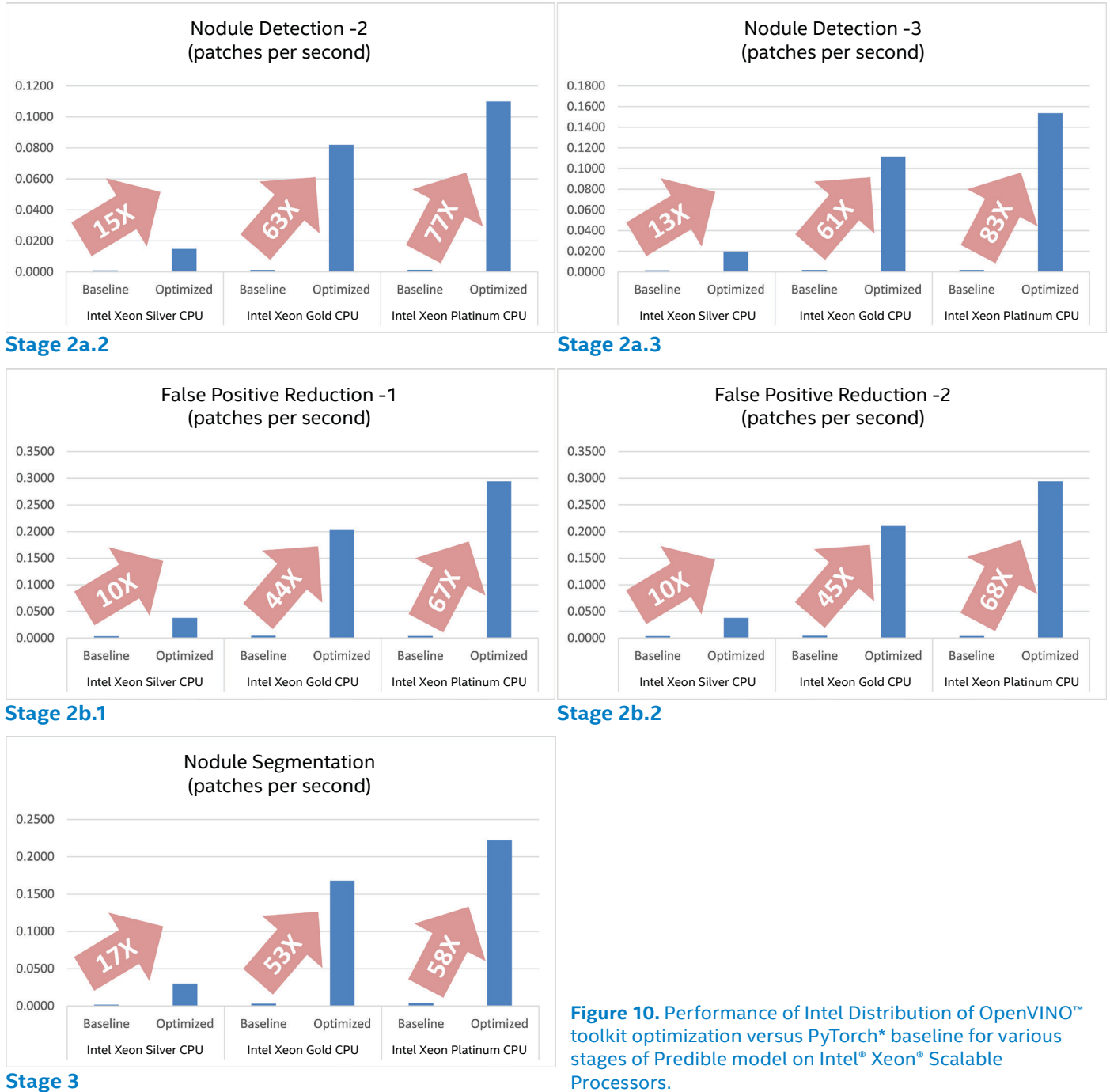
Table 2 depicts the number of layers per module in the Predible model before and after optimizations, clearly indicating the power of the toolkit. Performance improvements for various stages of the Predible model are dependent on the CPU used.

Modules in Predible Lung CT model	Baseline (PyTorch)	Optimized (OpenVINO)
Stage 1 – Lobe Segmentation	62	46
Stage 2a.1 – Nodule Detection	138	69
Stage 2a.2 – Nodule Detection	138	69
Stage 2a.3 – Nodule Detection	111	61
Stage 2b.1 – False Positive Reduction	30	20
Stage 2b.2 – False Positive Reduction	32	20
Stage 3 – Nodule Segmentation	61	35

Table 2. Number of layers per module before and after optimizations.

Table 3 describes the number of patches per stage of the Lung CT model and the overall time taken for the completion of each stage using the chosen Intel® Xeon® processors. Total time taken for all modules is under a minute for Intel® Xeon® Gold and Platinum processors and under five minutes for Intel® Xeon® Silver processors, clearly highlighting tradeoffs between latency and cost. For dedicated compute assets executing only one inferencing model, it may make sense to pick an entry processor that meets throughput needs while keeping costs low. For shared compute assets, where several inferencing models are executed concurrently and may have differing throughput and latency needs, Gold and Platinum parts offer the needed compute power and the agility to handle concurrent invocations of models with differing performance needs. Also depicted in Table 3 is the choice of instance and batch size for each module in various stages of the Predible Lung CT model that provides the best performance on different processors.

Modules		Number of Patches (p)	Intel® Xeon® Scalable Processor Family (Intel® Xeon® Silver 4110 Processor / Intel® Xeon® Gold 6140 Processor / Intel® Xeon® Platinum 8168 Processor) [1.1 / 5.3 / 8.3] TFLOPS											
			Silver			Gold			Platinum			Overall time (secs.)		
Stage #	Function		Instance (i)	Batch size (b)	Time taken per (i*b) [sec]	Instance (i)	Batch size (b)	Time taken per (i*b) [sec]	Instance (i)	Batch size (b)	Time taken per (i*b) [sec]	Silver	Gold	Platinum
1	Lobe Segmentation	324	8	1	0.4	6	1	0.12	6	1	0.1	16.4	6.5	5.4
2a	Nodule Detection -1	84	4	1	3.2	6	1	0.87	12	1	1.3	67.2	12.2	9.1
	Nodule Detection -2	84	4	1	3.2	6	1	0.87	12	1	1.3	67.2	12.2	9.1
	Nodule Detection -3	84	4	1	2.41	6	1	0.64	4	1	0.31	50.6	9.0	6.5
2b	False Positive Reduction -1	100	4	1	1.05	6	1	0.29	6	1	0.2	26.3	4.9	3.4
	False Positive Reduction -2	100	4	1	1.05	6	1	0.28	6	1	0.2	26.3	4.8	3.4
3	Nodule Segmentation	30	2	1	2.22	6	1	1.19	2	1	0.3	33.3	6.0	4.5
									Total time for all modules (secs.)			287.2	55.4	41.4

Table 3. Time taken per module in each stage of the Predible Lung CT model.

Conclusion

Intel and Predible Health researchers realized three main conclusions from their work:

- Intel Distribution of OpenVINO® toolkit-based optimizations yield significant speed ups (up to 83X) on Intel® Xeon® Scalable processors vs. baseline configurations.

- Intel Xeon Scalable processors offer a range of performance/price options to meet a variety of workload needs.

- Predible Health's complex Lung CT model can be computed in less than a minute of processing time.

References

1 Global Burden of Disease Collaborative Network. "Global Burden of Disease Study 2017 (GBD 2017) Results," Institute for Health Metrics and Evaluation (IHME), Seattle, United States, 2018.

2 N. Howlader, A. Noone, M. Krapcho, D. Miller, A. Brest, M. Yu, J. Ruhl, Z. Tatalovich, A. Mariotto, D. Lewis, D. Lewis, E. Feuer and K. Cronin, "SEER Cancer Statistics Review," National Cancer Institute, Bethesda, MD, 1975-2016.

3 "Reduced Lung-Cancer Mortality with Low-Dose Computed Tomographic Screening," New England Journal of Medicine, vol. 365, no. 5, pp. 395-409, 2011.

4 V. Doria-Rose and E. Szabo, "Screening and Prevention of Lung Cancer," in Lung Cancer: A Multidisciplinary Approach to Diagnosis and Management, New York, 2011, pp. 53-72.

5 D. Naidich, C. Marshall, C. Gribbin, R. Arams and D. McCauley, "Low-dose CT of the lungs: preliminary observations," Radiology, pp. 729-731, 1990.

6 H. MacMohan, A. John H.M. and G. G, "Guidelines for management of small pulmonary nodules detected on CT scans: a statement from Fleischner Society," Radiology, vol. 237, no. 2, pp. 395-400, 2005.

7 https://www.mayoclinic.org/diseases-conditions/lung-cancer/expert-answers/lung-nodules/faq-20058445.

8 K. Prakashini, S. Babu, K. Rajgopal and K. Kokila, "Role of Computer Aided Diagnosis (CAD) in the detection of pulmonary nodules on 64 row multi detector computed tomography.," Lung India, vol. 33, no. 4, pp. 391-397, 2016.

9 B. Sahiner, "Effect of CAD on radiologists' detection of lung nodules on thoracic CT scans: analysis of an observer performance study by nodule size.," Academic Radiology, pp. 1518-1530, 2009.

10 A. A. A. Setio, A. Traverso, T. d. Bel, M. S. Berens, C. v. d. Bogaard, P. Cerello, H. Chen, Q. Dou, M. E. Fantacci, B. Geurts, R. v. d. Gugten, P. A. Heng, B. Jansen and M. M. d. Kaste, "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge," CoRR, vol. abs/1612.08012, 2016.

11 F. Liao, M. Liang, Z. Li, X. Hu and S. Song, "Evaluate the Malignancy of Pulmonary Nodules Using the 3D Deep Leaky Noisy-or Network," coRR, vol. abs/1711.08324, 2017.

12 Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," CoRR, vol. abs/1606.06650, 2016.

13 S. G. Armato III, "The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A Completed Reference Database of Lung Nodules on CT Scans," Medical Physics, vol. 38, no. 2, pp. 915-931, 2011.

14 S. Zagoruyko and N. Komodakis, "Wide Residual Networks," CoRR, 2016.

15 T. Alexander, "Practice Policy and Quality Initiatives: Decreasing Variability in Turnaround Time for Radiographic Studies from the Emergency Department.," RadioGraphics, vol. 33, no. 2, pp. 361-371, 2013.

16 B. Bartholmai, "The Lung Tissue Research Consortium: An extensive open database containing histological, clinical, and radiological data to study chronic lung disease.," Insight Journal.

Appendix A: Hardware and Software Test Configurations

	Platinum	Gold	Silver
Tested By	Intel	Intel	Intel
Test Date	02-07-2019	02-07-2019	02-07-2019
Platform	S2600STQ	S2600WFQ	S2600BPB
#Nodes	1	1	1
#Sockets	2	2	2
CPU	8168	6140	4110
Cores per socket/Threads per socket	24/24	18/18	8/8
Serial No cpu0	-	-	-
Serial No cpu1	-	-	-
ucode	0x200005e	0x200005e	0x200005e
HT	off	off	off
Turbo	off	off	off
BIOS Version	SE5C620.86B.00.01.0 009.101920170742	SE5C620.86B.00.01.0 009.101920170742	SE5C620.86B.00.01.0 015.110720180833
System DDR Mem Config: Slots/Cap/Run-speed	12/16GB/2666	12/16GB/2666	12/16GB/2666
System DCPMM Config: Slots/Cap/Run-speed	-	-	-
Total Memory/Node	192GB	192GB	192GB
Storage-Boot	INTEL SSDSC2KB48 480GB	INTEL SSDSC2KB48 480GB	INTEL SSDSC2KB96 960GB
Storage-Application	-	-	-
NIC	-	-	-
PCH	-	-	-
Other HW (Accelerator)	-	-	-
OS	Ubuntu 16.04.6 LTS	Ubuntu 16.04.6 LTS	Ubuntu 16.04.6 LTS
Kernel	GNU/Linux 5.1.5-050105-generic x86_64	GNU/Linux 5.1.5-050105-generic x86_65	GNU/Linux 5.1.5-050105-generic x86_66
Mitigation Variants	Mitigated	Mitigated	Mitigated

Workload & Version	Stage 1	Stage 2a.1	Stage 2a.2	Stage 2a.3	Stage 2b.1	Stage 2b.2	Stage 3
Compiler	5.4.0	5.4.0	5.4.0	5.4.0	5.4.0	5.4.0	5.4.0
Libraries	MKL-DNN (OpenVINO™ inbuilt version)	MKL-DNN (OpenVINO™ inbuilt version)	MKL-DNN (OpenVINO™ inbuilt version)	MKL-DNN (OpenVINO™ inbuilt version)	MKL-DNN (OpenVINO™ inbuilt version)	MKL-DNN (OpenVINO™ inbuilt version)	MKL-DNN (OpenVINO™ inbuilt version)
Frameworks Version	OpenVINO™ 2019 R1	OpenVINO™ 2019 R1	OpenVINO™ 2019 R1	OpenVINO™ 2019 R1	OpenVINO™ 2019 R1	OpenVINO™ 2019 R1	OpenVINO™ 2019 R1
Dataset	PH dataset	PH dataset	PH dataset	PH dataset	PH dataset	PH dataset	PH dataset
Topology	3D Unet	3D Resnet	3D Resnet	3D Resnet	3D Wide Resnet	3D Wide Resnet	3D Unet
Batch Size	1	1	1	1	1	1	1