Creating World Changing Technologies

Creating
World
Changing
Technologies

Accelerating Deep Learning with Intel Extension for PyTorch: a MedMNIST classification decathlon example

Dr. Severine Habert
AI Software Solutions Engineer

intel.

Intel® AI Analytics Toolkit

Powered by oneAPI

Accelerate end-to-end AI and data analytics pipelines with libraries optimized for Intel® architectures

Who Uses It?

Data scientists, AI researchers, ML and DL developers, AI application developers

Top Features/Benefits

- Deep learning performance for training and inference with Intel optimized DL frameworks and tools
- Drop-in acceleration for data analytics and machine learning workflows with compute-intensive Python packages

Deep Learning

- Intel® Optimization for TensorFlow
- Intel® Optimization for PyTorch
- Intel® Neural Compressor
- Model Zoo for Intel® Architecture

Data Analytics & Machine Learning

Accelerated Data Frames

- Intel® Distribution of Modin
- OmniSci Backend

Intel® Distribution for Python

- XGBoost
- Scikit-learn
- Daal-4Py
- NumPy
- SciPy
- Pandas

Samples and End2End Workloads

CPU GPU

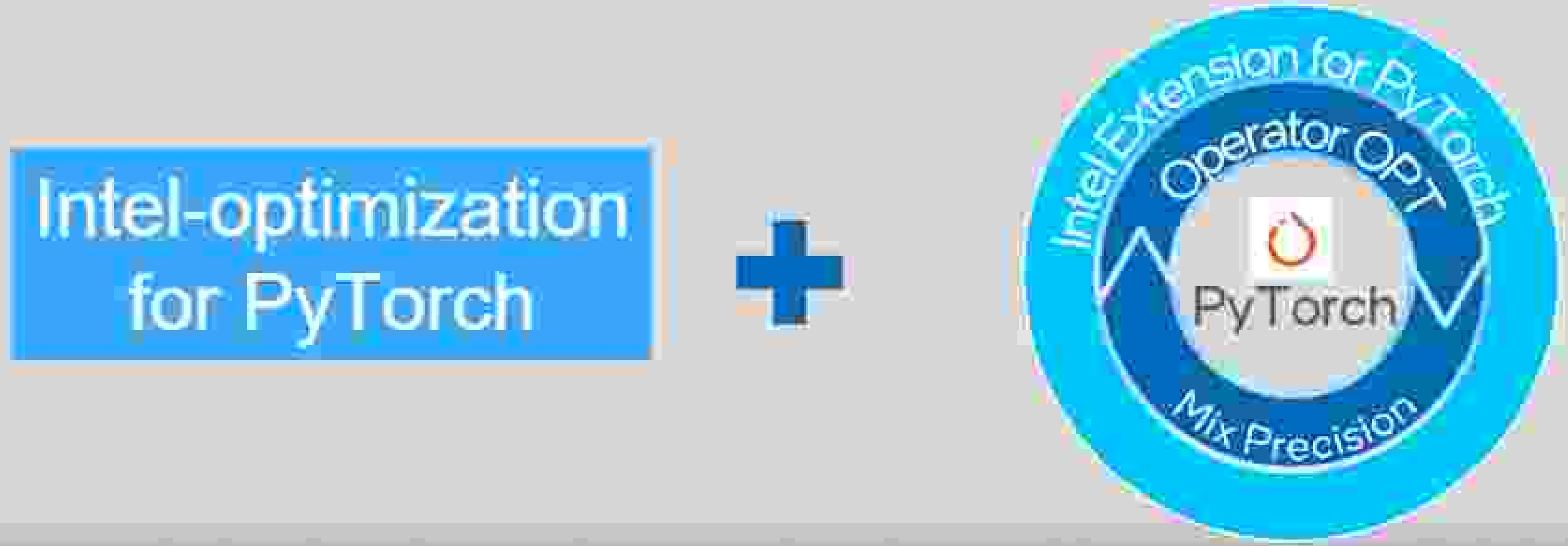
Supported Hardware Architectures[1]

Get the Toolkit HERE or via these locations

Intel Installer | Docker | Apt, Yum | Conda | Intel® DevCloud

Intel-optimization for PyTorch and Intel Extension for PyTorch

- Intel-optimized PyTorch is the version upstreamed to stock PyTorch, optimized by OneDNN in the backend
- Intel Extension for PyTorch (IPEX) is an additional module for functions not supported in standard PyTorch (such as mixed precision).



How to activate Intel Extension for PyTorch?

Version <10 as in Intel OneAPI Analytics Toolkit 2021.4

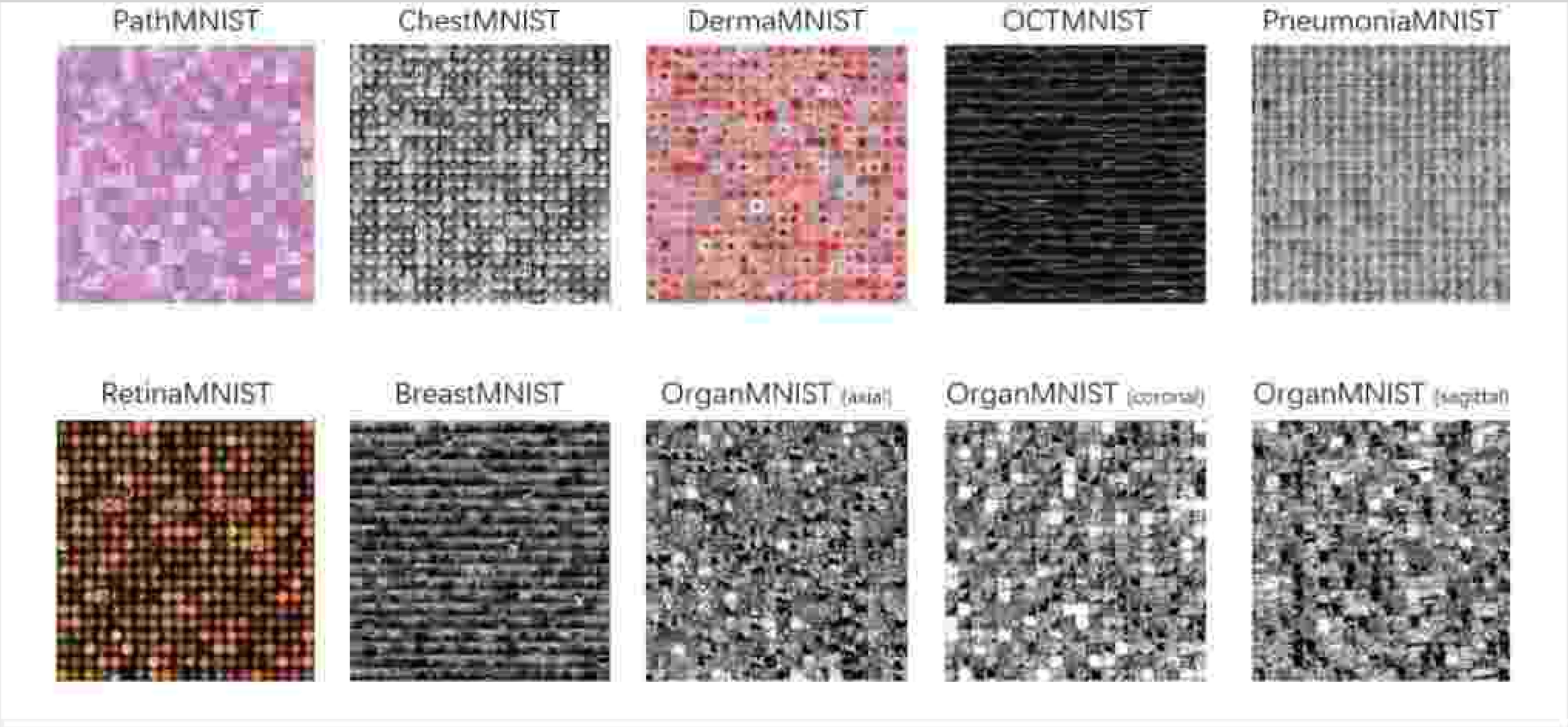
- import intel_pytorch_extension as ipex
- device = torch.device("xpu")

Version == 10 as in Github

- import intel_extension_for_pytorch as ipex
- model = ipex.optimize(model)

Only 2 lines code change in both cases

Datasets



Jiancheng Yang, Rui Shi, Bingbing Ni. "MedMNIST Classification Decathlon: A Lightweight AutoML Benchmark for Medical Image Analysis." , IEEE 18th International Symposium on Biomedical Imaging (ISBI) , p 191-195,2021

intel.

Demo - training

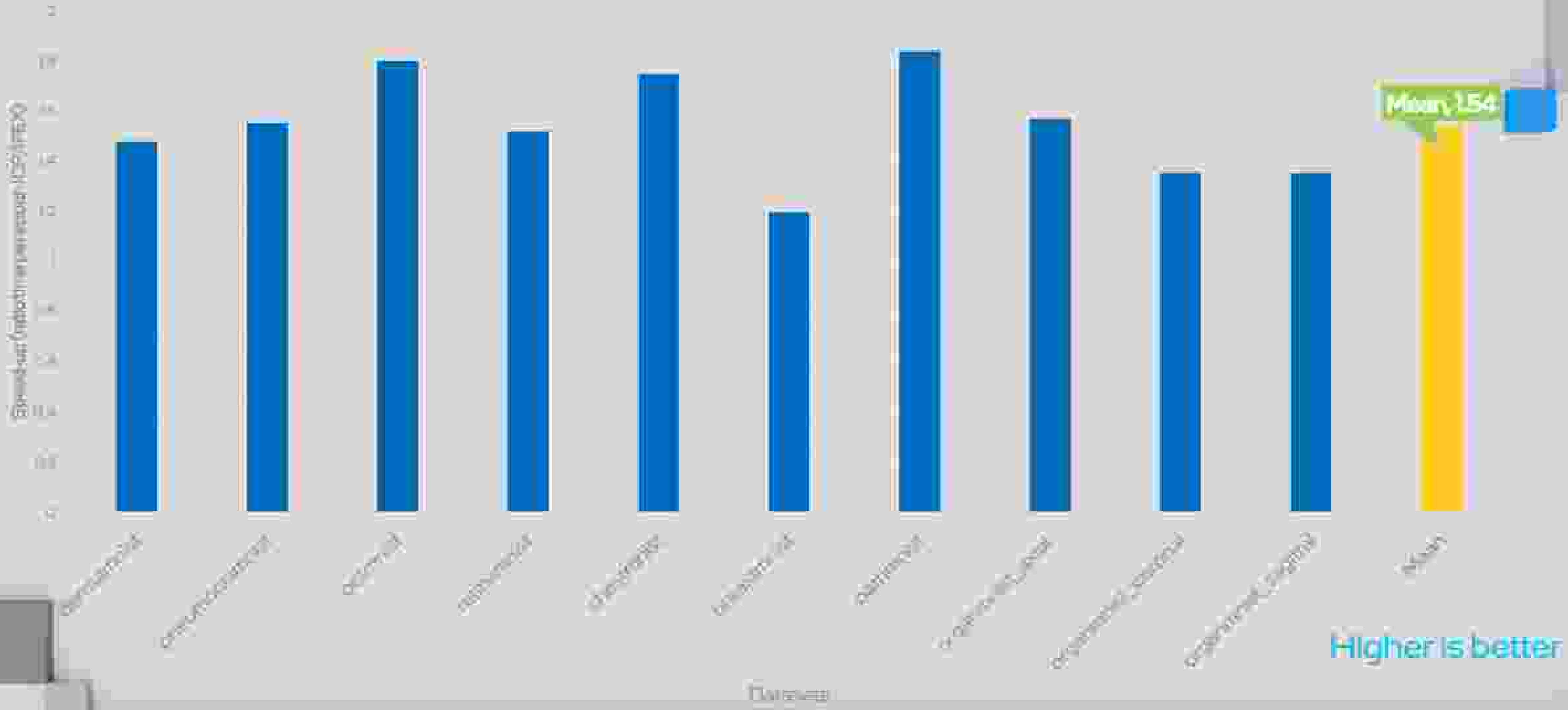
- Switching to Jupyter notebook for showcasing demo
- training_ISC-presentation slides

```
In [ ]:  for epoch in trange(0,nb_epochs):
             time1 = time.time()
             train(model, optimizer, criterion, train_loader, device, task)
             time_epoch = time.time()-time1
             print("training "+str(time_epoch)+ "sec")
             if nb_epochs>0: # avoid the warm-up effect
                 time_training.append(time_epoch)
             val(model, val_loader, device, val_acc_list, task, dir_path, epoch)

         average_training_time_per_epoch=mean(time_training)
         print("Training per epoch time "+str(average_training_time_per_epoch) + "s")
```

intel

Training Speed-up Intel Extension for PyTorch (IPEX) vs Intel-optimized PyTorch (IOP)

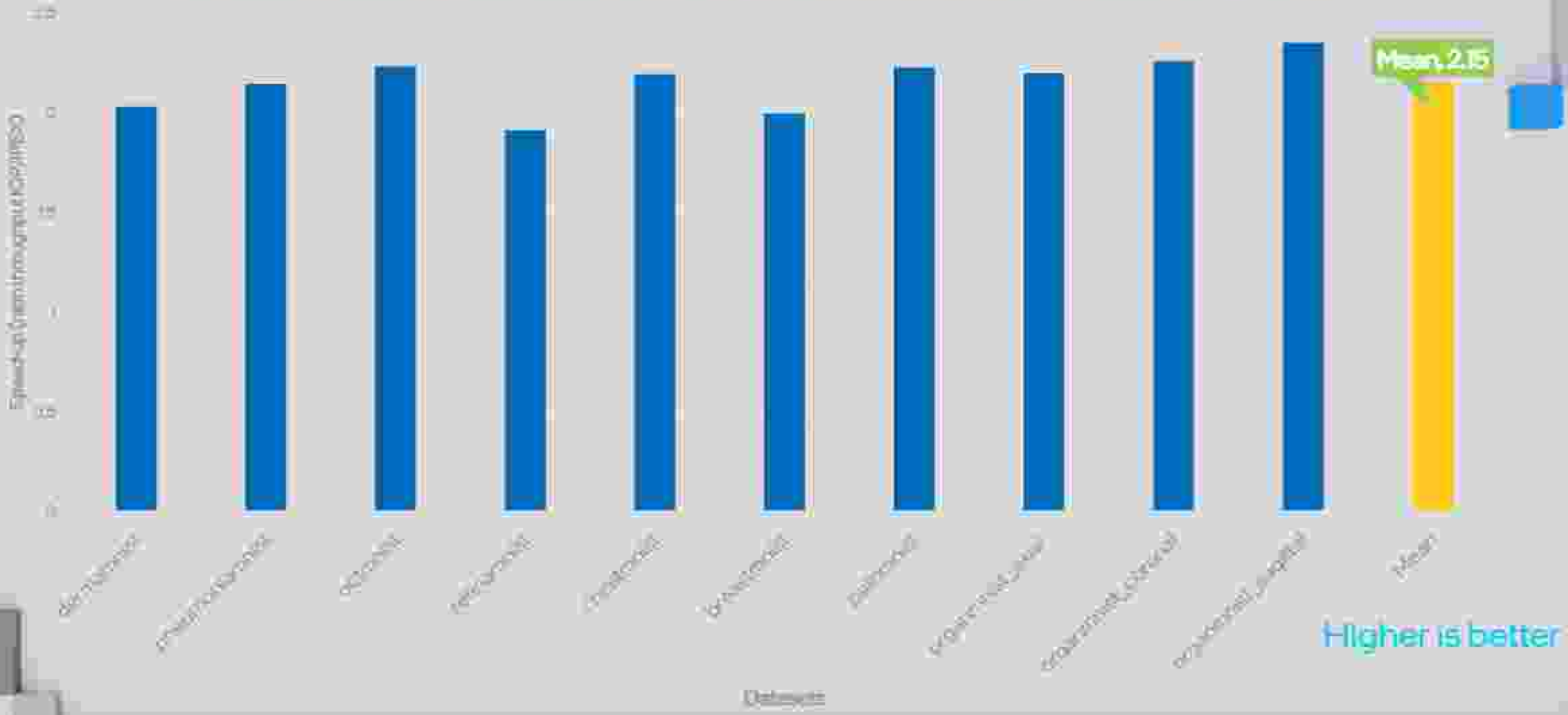


See backup for workloads and configurations. Results may vary.

Demo - inference

- Switching to Jupyter notebook for showcasing demo
- inference_ISC slides

Inference Speed-up Intel® Extension for PyTorch (IPEX) vs Intel®-optimized PyTorch



See backup for workloads and configurations. Results may vary.

intel.

Call for action

Find resources in the links below

Demo Github

Scan Me

oneAPI AI Analytics Toolkit

Scan Me

Intel Extension for PyTorch Github

Scan Me

Intel DevCloud for oneAPI

Scan Me

Notices & Disclaimers

- Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.
- Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.
- Your costs and results may vary.
- Intel technologies may require enabled hardware, software or service activation.
-

Appendix

- **Workload:** Training Speed-up Intel® Extension for PyTorch (IPEX) vs Intel®-optimized PyTorch (IOP) - ratio time per epoch IOP/ time per epoch IPEX) for ResNet18 on 10 datasets (datasets: https://zenodo.org/record/4269852)
- Test Intel-optimized PyTorch (IOP): 1-node, 2x Intel Xeon Platinum 8360Y processor on Coyote Pass with 512 GB (16 slots/ 32GB/ 3200) total DDR4 memory, ucode 0xd0001e0, HT on, Turbo on, Ubuntu 20.04.01 LTS, 5.10.5-1x intel_SSDSC2KG96, ResNet18, gcc-9.3.0, oneDNN 1.6.4, Intel-optimized PyTorch 1.7 from oneAPI AI Analytics Toolkit 2021.2, F32, BS see below, test by Intel on May 11th, 2021
- Test Intel-extension for PyTorch (IPEX): 1-node, 2x Intel Xeon Platinum 8360Y processor on Coyote Pass with 512 GB (16 slots/ 32GB/ 3200) total DDR4 memory, ucode 0xd0001e0, HT on, Turbo on, Ubuntu 20.04.01 LTS, 5.10.5-1x intel_SSDSC2KG96, ResNet18, gcc-9.3.0, oneDNN 1.6.4, Intel Extension for PyTorch 1.2.0 from oneAPI AI Analytics Toolkit 2021.2, FP32, BS see below, test by Intel on May 11th, 2021

	BS training
dermamnist	1024
pneumoniamnist	1024
octmnist	8192
retinamnist	1024
chestmnist	4096
breastmnist	1024
pathmnist	8192
organmnist_axial	2048
organmnist_coronal	1024
organmnist_sagittal	1024

Appendix

- **Workload:** Inference Speed-up Intel® Extension for PyTorch (IPEX) vs Intel®-optimized PyTorch (IOP) - ratio throughput IOP/ throughput IPEX) for ResNet18 on 10 datasets (datasets: https://zenodo.org/record/4269852)
- Test intel-optimized PyTorch (IOP): 1-node, 2x Intel Xeon Platinum 8360Y processor on Coyote Pass with 512 GB (16 slots/ 32GB/ 3200) total DDR4 memory, ucode 0xd0001e0, HT on, Turbo on, Ubuntu 20.04.01 LTS, 5.10.5-1x intel_SSDSC2KG96, ResNet18, gcc-9.3.0, oneDNN 1.6.4, Intel-optimized PyTorch 1.7 from oneAPI AI Analytics Toolkit 2021.2, F32, BS see below, test by Intel on May 11th, 2021
- Test Intel-extension for PyTorch (IPEX): 1-node, 2x Intel Xeon Platinum 8360Y processor on Coyote Pass with 512 GB (16 slots/ 32GB/ 3200) total DDR4 memory, ucode 0xd0001e0, HT on, Turbo on, Ubuntu 20.04.01 LTS, 5.10.5-1x intel_SSDSC2KG96, ResNet18, gcc-9.3.0, oneDNN 1.6.4, Intel Extension for PyTorch 1.2.0 from oneAPI AI Analytics Toolkit 2021.2, FP32, BS see below, test by Intel on May 11th, 2021

	BS inference
dermamnist	7007
pneumoniamnist	4708
octmnist	97477
retinamnist	1080
chestmnist	78468
breastmnist	546
pathmnist	89996
organmnist_axial	34581
organmnist_coronal	13000
organmnist_sagittal	13940

intel.

Dataset acknowledgement

PathMNIST: Jakob Nikolas Kather, Johannes Krisam, et al., "Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study," PLOS Medicine, vol. 16, no. 1, pp. 1–22, 01 2019.
License: CC BY 4.0

ChestMNIST: Xiaosong Wang, Yifan Peng, et al., "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in CVPR, 2017, pp. 3462–3471.
License: CC0 1.0

DermaMNIST:
- Philipp Tschandl, Cliff Rosendahl, and Harald Kittler, "The ham10000 dataset, a large collection of multisource dermatoscopic images of common pigmented skin lesions," Scientific data, vol. 5, pp. 180161, 2018.
- Noel Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, Harald Kittler, and Allan Halpern: "Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)", 2018; arXiv:1902.03368.

License: CC BY-NC 4.0

OCTMNIST/PneumoniaMNIST: Daniel S. Kermany, Michael Goldbaum, et al., "Identifying medical diagnoses and treatable diseases by image-based deep learning," Cell, vol. 172, no. 5, pp. 1122 – 1131.e9, 2018.
License: CC BY 4.0

RetinaMNIST: DeepDR Diabetic Retinopathy Image Dataset (DeepDRiD), "The 2nd diabetic retinopathy – grading and image quality estimation challenge," https://isbi.deepdr.org/data.html, 2020.
License: CC BY 4.0

BreastMNIST: Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy, "Dataset of breast ultrasound images," Data in Brief, vol. 28, pp. 104863, 2020.
License: CC BY 4.0

OrganMNIST_{Axial,Coronal,Sagittal}:
Patrick Bilic, Patrick Ferdinand Christ, et al., "The liver tumor segmentation benchmark (lits)," arXiv preprint arXiv:1901.04056, 2019.
Xuanang Xu, Fugen Zhou, et al., "Efficient multiple organ localization in ct image using 3d region proposal network," IEEE Transactions on Medical Imaging, vol. 38, no. 8, pp. 1885–1898, 2019.
License: CC BY 4.0

intel.

intel®