PEGATRON
和 碩 聯 合 科 技

PEGATRON OpenVINO Quantization White Paper

PEGATRON
和碩聯合科技

Contents

01 Purpose

A deep learning model usually has tens or hundreds of millions of parameters. To process these parameters requires high computing power and memory. It is very important to reduce the model size for higher efficiency in inference task.

Quantization is one of the methods to reduce model size. The core idea is to optimize a model by representing model parameters with low-precision data types (such as INT8 and FP16) without incurring a significant accuracy loss. In this paper, PEGATRON verifies this tool via face recognition model created through Intel OpenVINO.

Intel OpenVINO is a comprehensive toolkit that user can use to develop and deploy vision-oriented solution on Intel platforms. It offers an optimization tool which is for model quantization and accuracy checker to check the accuracy of the different models.

PEGATRON face recognition model has been developed for smart surveillance. The model is based on state-of-the-art face recognition model (ArcFace +MS1MV2+R100) as Figure 1, and we retrain it on Pegatoncorp's face dataset for the requirements.

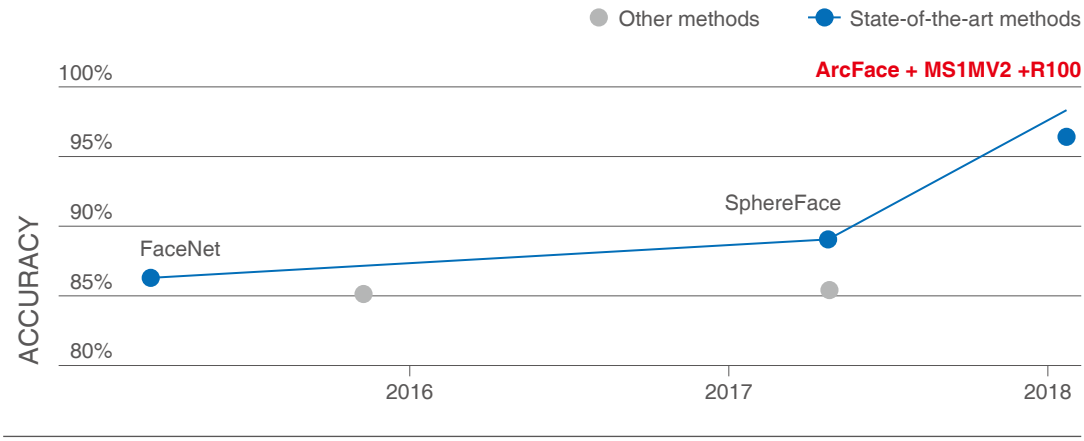


Figure 1.

MegaFace is a large-scale public face recognition training dataset that serves as one of the most important benchmarks for commercial face recognition vendors. ArcFace achieves state-of-the-art results on the MegaFace Challenge.

The model can achieves 326FPS throughput by applying existing solution without VNNI, it means we can handle only 8 IP cameras (with HD 1080p resolution) for 6 FPS requirements

If we want to increase the throughput to handle more video inputs, the most intuitive way is to increase the number of GPU (as Figure 2) or re-design model. However, it requires extra cost and time for the improvement.

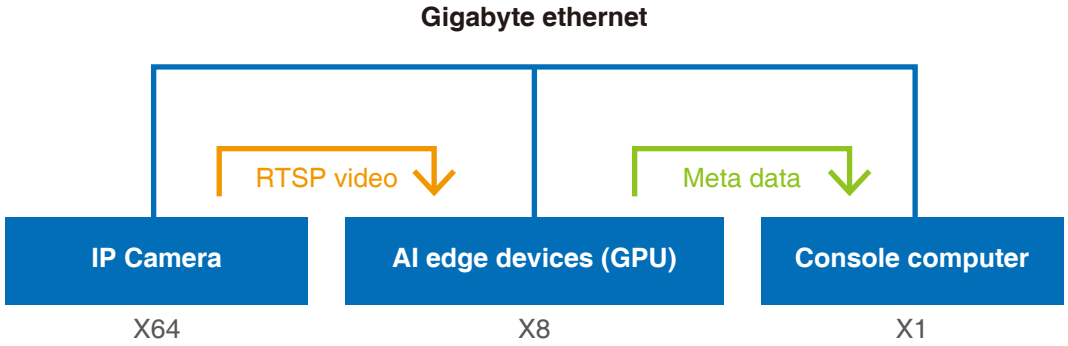


Figure 2.

Based on the current solution, smart surveillance need 8 GPUs to handle 64 IP cameras the requirements.

Therefore, PEGATRON cooperates with Intel & ITRI, to use OpenVINO Quantization technology to increase the throughput of AI model in a more efficient way.

02 Environments

Intel Deep Learning Boost(DL Boost) is designed to deliver significant, more efficient model acceleration, and AVX-512 extension Instruction Sets VNNI (Vector Neural Network Instructions) is one realization of DL Boost for INT8 inference on 2nd Gen Intel® Xeon® Scalable processors.

VNNI uses a single instruction for deep learning computations(matrix multiplication operations) that formerly required three separate instructions, as Figure 3.

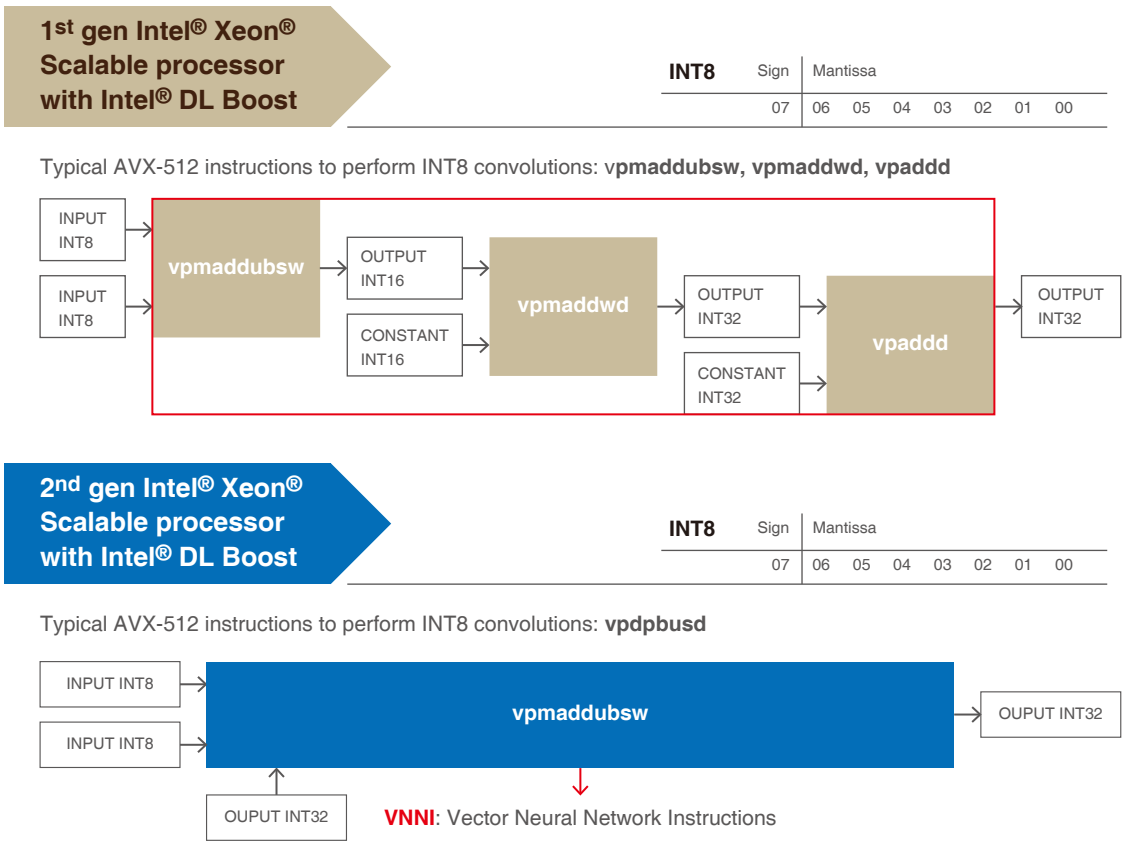


Figure 3.

In 2nd generation Intel® Xeon® Scalable processors, convolutions in Intel MKL-DNN occur in INT8 precision via one Intel AVX-512 vpdpbusd instruction. Since the low-precision operation now uses a single instruction, two of these instructions can be executed in a given cycle. Reduced precision and use of a single instruction optimizes utilization of the microarchitecture for each convolution operation in a neural network and brings significant performance benefits.

With VNNI, we expects it can improve the number of elements processed per cycle by 2x and 3x respectively. Therefore, we need to use Intel® Xeon® Scalable processors for accelerating the INT8 inference throughput and reducing the latency.

2.1 Hardware

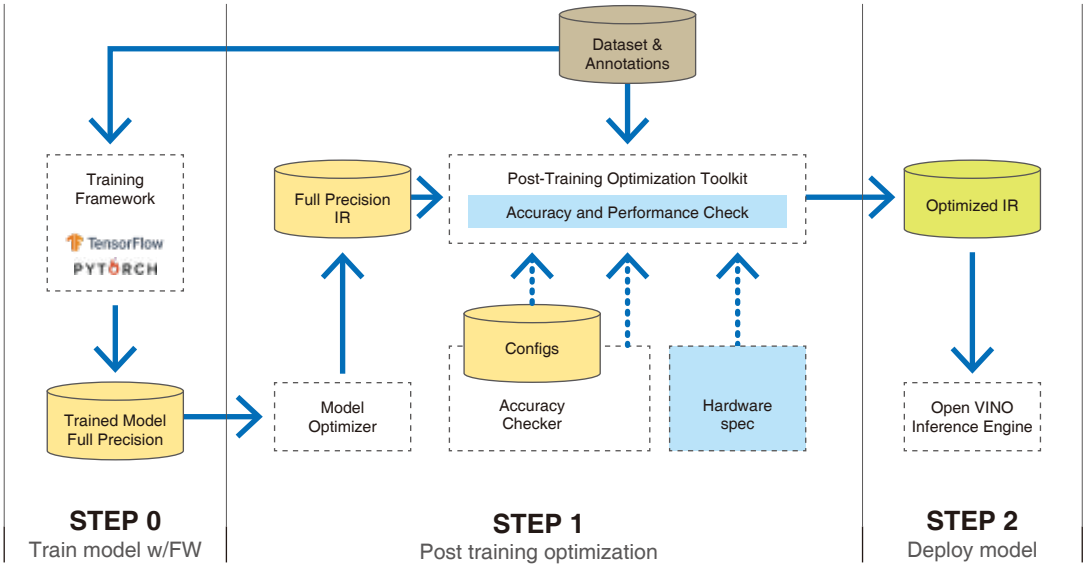
CPU	Intel® Xeon® Gold 6240R CPU @ 2.40GHz
Number of Sockets	2
CPU cores per socket	24
CPU cores in total	48
Memory Model	Micron DIMM DDR4 Synchronous 2933 MHz
Memory	188G

2.2 Software

OS	Ubuntu 18.04.1 LTS
Kernel Version	4.15.0-29-generic
OpenVino Version	2020.3 LTS

03 Quantization

IThe figure below depicts the system architecture and components in OpenVINO Quantization steps.



STEP 0

PEGATRON has developed face recognition model and train model with our dataset based on certain MxNet framework in FP32 precision.

STEP 1

IR is the representation of a model used by OpenVINO, we use **Model Optimizer** tool to convert the pre-trained model to IR format as below command:

```
$python3 ${OPNVINO_HOME}/deployment_tools/model_optimizer/mo_mxnet.py --input_model model-0001.params
```

Then, we use **Post Training Optimization(PTO)** tool to quantize mode as below command:

```
$ python3 ${OPNVINO_HOME}/deployment_tools/tools/post_training_optimization_toolkit/main.py -c pega_arcface_r100_default_quantile.json
```

The PTO will leverage accuracy checker to evaluate the IR model for controlling the drop in the accuracy during quantization. Finally, we can use accuracy checker to verify the accuracy is reliable in the other benchmark dataset (LFW), and benchmark tool to evaluate the performance.

Accuracy checker command:

```
$ python3 ${OPNVINO_HOME}/deployment_tools/open_model_zoo/tools/accuracy_checker -c
pega_arcface_r100_int8.yml
```

Benchmark command:

```
$ python3 ${OPNVINO_HOME}/deployment_tools/open_model_zoo/tools/bench
mark_tool/benchmark_app.py -c pega_arcface_r100_int8.xml
```

STEP 2

The INT8 quantized model is ready to be deployed on target platform. It's worthwhile to mention that the quantization is hardware-specific. It means that if user quantizes a model on machine A and want to deploy the quantized model on machine B, the machine A and machine B should have the same ISA.

04 Results

As a below table, Pegatron face recognition model inference throughput faster than original solution without VNNI(326 FPS) based on OpenVino Quantization and VNNI technology.

Device	Model Precision	Flops	LFW Accuracy	Throughput
Intel Xeon 8276 w/o VNNI	INT8	6.4G	99.63%	326 FPS
Intel Xeon 6240R w VNNI	INT8	6.4G	99.63%	579 FPS

Table 1.

Shows the comparison, INT8 quantized mode can improve the throughput and with VNNI the improvement may up to 1.77x with minimal accuracy loss (< 0.07%).

05 Summary

In this project, PEGATRON verifies Intel OpenVINO toolkit and how it can be used to improve our model performance quantization technology. We successfully converted PEGATRON face recognition model into INT8 precision, and it can achieve significant acceleration while maintaining accuracy as our expectation.

More important, lowering computational precision can be achieved with no model re-training or fine-tuning required and only requires a few steps. This enhanced pipeline reduces which is beneficial to deploy use cases.