About the OpenVINO™ Integration with TensorFlow

What are the Benefits for TensorFlow Developers?

How Is the Integration Used?

How Does it Work?

How is this Integration Different from OpenVINO?

How are Customers Adopting the OpenVINO Integration with TensorFlow?

How Does the OpenVINO Integration for TensorFlow

Compare with Nvidia TF-TRT?

Compatibility Questions

What Operating Systems, Hardware, and Frameworks are Supported?

Which Versions of TensorFlow are Supported?

Which Models are Supported?

Would this Flow Work with TensorFlow Models from

CustomVision.ai from Microsoft Azure?

Are TensorFlow Lite Models Supported?

Are Models with Dynamic Shapes Supported?

What if a Layer isn’t Supported on OpenVINO?

Is this Feature Open Source or Proprietary?

Performance Questions

How Well Does it Perform?

Does the Integration Increase Inference Latency?

Any Plans to Supporting/Integrating This to OpenVINO

Benchmarking App With TF Models?

Does the OpenVINO Integration with TensorFlow support Quantization?

Development Questions

Where Can I Get Sample Working Code?

Are C++ APIs Supported?

Using the openvino_tensorflow API, Can Developers Tune OpenVINO

Inference Engine Parameters, Such as the device_plugin Parameters?

Working with the OpenVINO toolkit

Can TensorFlow Models be Imported into the OpenVINO API

Without Using the Model Optimizer?

How can I Export a TensorFlow Model’s Intermediate Representation

(IIR) File from the OpenVINO Integration with TensorFlow?

Do Developers Have to Update Application Code When Switching Between

the OpenVINO Toolkit and the OpenVINO Integration with TensorFlow?

Installation Options and Dependencies

What are we Releasing?

Where Can I Find all the Dependencies Packages Used to Build the Source?

Timeline and Future Developments

What’s the Timeline for the Next Release of the OpenVINO Integration with

TensorFlow?

Are we Planning a Similar Integration with PyTorch or PaddlePaddle?

Find Out More

Where Can I Go for Further Questions?

Other questions? Email openvino-tensorflow@intel.com
About the OpenVINO™ Integration with TensorFlow

What Are the Benefits for TensorFlow Developers?

The OpenVINO™ integration with TensorFlow delivers OpenVINO™ toolkit inline optimizations and the OpenVINO runtime with enhanced TensorFlow compatibility. The integration is designed for developers who would like to boost performance of their inferencing applications with minimal code modifications. It accelerates inference across many AI models on a range of Intel® processors such as:

- Intel® CPUs. For example, Intel® Core™ processors and Intel® Xeon® processors. These include the newer Intel® Xeon® processor SKUs, such as 2nd gen Intel® Xeon® Scalable and 11th gen Intel® Core™ Processors.
- Intel® integrated GPUs
- Intel® Movidius™ Vision Processing Units (VPUs)
- Intel® Vision Accelerator Design with 8 Intel® Movidius™ Myriad™ X VPUs (VAD-M or HDDL, short for High-Density Deep Learning)

Developers using this integration benefit from the following advantages:

- Performance acceleration compared to native TensorFlow (depending on the underlying hardware configuration). We’ll publish official benchmarking results soon. See customer results following.
- Accuracy nearly identical to the original model.
- Simplicity. Developers continue to use TensorFlow APIs for inferencing. No need to refactor code. Just import, enable, and set device.
- Robustness. The solution supports a range of TensorFlow models and operators, across a variety of OS/Python environments.
- Inline model conversions. No explicit model conversion is required.
- Minimal incremental memory and disk footprint required.
- Support for a broad range of Intel processor powered devices—CPUs, iGPUs, VPUs, HDDL.
**How Is the Integration Used?**

To accelerate TensorFlow performance with the integrated OpenVINO toolkit, add these two lines to your Python code or Jupyter notebook (see Figure 1):

```python
import openvino_tensorflow
openvino_tensorflow.set_backend('<backend_name>')
```

Supported backends are “CPU”, “GPU”, “GPU_FP16”, “MYRIAD”, and “VAD-M”. You can only use one backend at a time in the OpenVINO integration with TensorFlow. The full Intel® Distribution of OpenVINO™ toolkit enables multiple different backends to be used simultaneously.

CPU is the default backend target, so if the inferencing is to run on CPUs, setting the backend is optional. For other hardware targets, it is required.

**How Does it Work?**

The OpenVINO integration with TensorFlow provides accelerated TensorFlow performance by efficiently partitioning TensorFlow graphs into multiple subgraphs. These are then dispatched to either the TensorFlow runtime or the OpenVINO runtime for accelerated inferencing. The results are assembled to provide the final inference results (see Figure 2).

---

**Figure 1.** Code snippet showing how easy it’s to use different hardware with the openvino_tensorflow package.

**Figure 2.** End-to-end overview of the workflow.
Figure 3 shows a detailed architecture model. Find information about the components of this architecture here.

**How Is this Integration Different from OpenVINO?**

The OpenVINO integration with TensorFlow is designed for developers who want to use the OpenVINO toolkit to enhance inferencing performance with minimal code modifications.

For maximum performance, efficiency, tooling customization, and hardware control, we recommend adopting the full Intel® Distribution of OpenVINO™ toolkit, using the OpenVINO APIs and the native OpenVINO runtime.

The OpenVINO integration with TensorFlow is separate to the established OpenVINO workflow. With the release of this integration, there’s no change to the OpenVINO toolkit. This integration provides an additional way for developers to use OpenVINO.

Here’s the external product positioning:

<table>
<thead>
<tr>
<th>Offer</th>
<th>Benefits</th>
<th>Who is it For?</th>
<th>Who is it Not For?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native OpenVINO™ toolkit:</td>
<td>• Optimized performance, minimum memory &amp; disk usage</td>
<td>All developers who want optimized inferencing in deployment on Intel XPUs and are willing to learn and adopt the OpenVINO toolkit</td>
<td>• Developers with edge case models that don’t convert with MO</td>
</tr>
<tr>
<td>• Model Optimizer (MO) conversion to IR file</td>
<td>• Easy to use built-in optimization (quantization, graph optimization, XPU routing, etc)</td>
<td></td>
<td>• Data scientists focused on training (use Intel optimized frameworks)</td>
</tr>
<tr>
<td>• OpenVINO APIs with OpenVINO Runtime</td>
<td>• Access Intel® XPUs without rewriting (“write once, deploy anywhere”)</td>
<td></td>
<td>• Those who insist on maximum compatibility with framework APIs</td>
</tr>
<tr>
<td>• Associated tools (POT, NNCF, DL Workbench, etc)</td>
<td></td>
<td></td>
<td>• Traditional ML developers (use AI &amp; Analytics Toolkit)</td>
</tr>
<tr>
<td>OpenVINO™ Integration with TensorFlow (OV-TF) - add OpenVINO backend to TensorFlow with two lines of code</td>
<td>Get started with OpenVINO while maintaining maximum API &amp; model compatibility with TensorFlow</td>
<td>Developers • Exploring OpenVINO • With models that aren’t yet supported • Who insist on maintaining API compatibility at the expense of some performance</td>
<td>All developers who want optimized inferencing performance in deployment on Intel XPUs</td>
</tr>
</tbody>
</table>

© Intel Corporation 2020. All rights reserved.
The following table is provided to help developers choose between the two options.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>OpenVINO™ toolkit (with Model Optimizer)</th>
<th>OpenVINO™ integration with TensorFlow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Faster</td>
<td>Slower (model-dependent)</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Same</td>
<td></td>
</tr>
<tr>
<td>Ease of use</td>
<td>Needs model conversion</td>
<td>No explicit model conversion required</td>
</tr>
<tr>
<td>Model conversion</td>
<td>Offline</td>
<td>Inline</td>
</tr>
<tr>
<td>Intel Hardware</td>
<td>All</td>
<td>No support for Field Programmable Gate Arrays (FPGAs)</td>
</tr>
<tr>
<td>INT4, INT2, Sparsity</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>OpenVINO™ Deep Learning workbench and other developer tools</td>
<td>Available</td>
<td>Not Available</td>
</tr>
<tr>
<td>Target Audience</td>
<td>OpenVINO developer</td>
<td>OpenVINO user</td>
</tr>
<tr>
<td>Expanded Developer Reach</td>
<td>Embedded developers, original device manufacturers (ODMs), Enterprise/cloud developers</td>
<td>Existing developers who prefer TensorFlow APIs</td>
</tr>
</tbody>
</table>
Q How are Customers Adopting the OpenVINO Integration with TensorFlow?

A Customers are using the OpenVINO integration with TensorFlow for a variety of use cases.

Extreme Vision’s CV MART is a dedicated cloud for AI. It offers developers a rich catalog of services, models, and frameworks to optimize their AI workloads on a variety of Intel® platforms, such as CPUs and iGPUs. The Extreme Vision team is testing the OpenVINO integration with TensorFlow with the goal of providing it to TensorFlow developers on CV MART. Extreme Vision tested various models for classification, object detection, instance segmentation, and 3D face reconstruction. The company saw significant geometric acceleration using the OpenVINO integration with TensorFlow*. Read Extreme Vision’s results here.

Terra is a more secure, open-source platform for biomedical researchers to access data, run analysis tools, and collaborate. The cloud-based platform is co-developed by the Broad Institute of MIT and Harvard, Microsoft, and Verily. The Terra platform includes Genome Analysis Toolkit (GATK) tools and pipelines. The Genome Analysis Toolkit (GATK), developed by the Broad Institute, is the world’s most widely used open-source toolkit for variant calling. CNNScoreVariants is one of the deep learning tools included in GATK, which applies a Convolutional Neural Net to filter annotated variants. In a blog, the Broad Institute showcases how to accelerate CNNScoreVariants without losing accuracy by using the OpenVINO integration with TensorFlow. The Broad Institute’s results showed a speed-up of 21 percent on an Intel Core i5 processor and 16.8 percent on an Intel Xeon processor**.

Q How Does the OpenVINO Integration for TensorFlow Compare with Nvidia TF-TRT?

A The solutions have similar functionality. Both enable TensorFlow developers to accelerate their models with just a couple of lines of code and online model conversion.
Compatibility Questions

Q What Operating Systems, Hardware, and Frameworks are Supported?
A We support Linux, Windows, and macOS. Please check this table for supported environments and installation instructions.

Q Which Versions of TensorFlow are Supported?
A TensorFlow versions 2.7.0, 2.5.1, and 2.4 are supported. For TF1.x models please email OpenVINO-tensorflow@intel.com.

Q Which Models are Supported?
A We support over 270 computer vision, NLP, and popular TFHub models, such as MASK R-CNN, BERT, and GPT2. Visit our model page for the latest list of models. This list is expanding as we continue to validate and add more.

Q Are TensorFlow Lite Models Supported?
A No, TensorFlow Lite models aren’t supported. If you have a specific customer/use-case, please contact us at OpenVINO-tensorflow@intel.com.

Q Are Models with Dynamic Shapes Supported?
A Yes, models with dynamic input shapes are supported.

Q Why is a Layer isn’t Supported on OpenVINO?
A If the layer is supported by OpenVINO, it is accelerated by the OpenVINO backend. Otherwise, it uses the TensorFlow runtime. If a layer works correctly in TensorFlow, it should work with this integration.

Q Is this Fallback Feature Open Source or Proprietary?
A It will be open sourced under an Apache 2.0 license.
Performance Questions

Q How Well Does it Perform?
A We’re benchmarking the OpenVINO integration with TensorFlow and will publish official results soon. One of our customers has published performance results based on its testing. Extreme Vision tested various models for classification, object detection, instance segmentation, and 3D face reconstruction. The company saw significant geomean acceleration using the OpenVINO integration with TensorFlow. We achieve close to OpenVINO API performance for models that are fully supported by OpenVINO native. Due to the additional overhead of subgraph partitioning, OCM, and clustering, OpenVINO integration with TensorFlow is expected to be somewhat slower than OpenVINO native performance. The tradeoff is that the OpenVINO integration with TensorFlow is easier to use and supports more models because unsupported TensorFlow operators are sent back to the native TensorFlow runtime for execution. See official OpenVINO performance benchmarks for OpenVINO toolkit performance results.

Q Where can we find OVTF performance numbers?
A We’ll be publishing performance numbers soon. But as stated earlier, the OpenVINO toolkit will have the best performance. Please refer to the officially published OpenVINO performance results.

Q Does the OpenVINO Integration with TensorFlow support Quantization?
A Yes. This feature is in Preview. The OpenVINO integration with TensorFlow supports TensorFlow models quantized with OpenVINO NNCF and TensorFlow tfmot. FP16 and INT8 quantization is supported at this time. We’re working on releasing additional code and documentation. Please contact us if you have a model that needs to be quantized and run.

Q Does the Integration Increase Inference Latency?
A No. Compared to native TensorFlow, the OpenVINO integration with TensorFlow accelerates models at lower latency. We’ve seen great performance in customer use cases. We are releasing performance numbers soon.
Development Questions

**Q** Where Can I Get Sample Code?

**A** The OpenVINO integration with TensorFlow works in a variety of environments—both in the cloud and at the edge. The underlying hardware needs to be an Intel platform.

For example, the integration works on the following cloud platforms:

- Intel® DevCloud for the Edge
- Google Colab [Classification example, Object Detection example]
- Amazon Web Services Deep Learning AMI. Ubuntu 18 & Ubuntu 20 on EC2 C5 instances optimized for inferencing
- Microsoft Azure ML

Any Intel® architecture-based edge device is supported.

Samples are available in the examples/ directory in the git repo.

**Q** Are C++ APIs Supported?

**A** Yes. Please refer to the C++ example in the documentation.

Using the openvino_tensorflow API, Can Developers Tune OpenVINO inference Engine Parameters, Such as the device_plugin Parameters?

The architecture allows for tunable parameters. Currently, we don’t expose any of them. If you have a specific customer or use case that requires such parameters, please contact us.

**Q** Using the openvino_tensorflow API, Can Developers Tune OpenVINO Inference Engine Parameters, Such as the device_plugin Parameters?

**A** The architecture allows for tunable parameters. Currently, we don’t expose any of them. If you have a specific customer or use case that requires such parameters, please contact us.
Working with the OpenVINO toolkit

Q How can I Export a TensorFlow Model’s Intermediate Representation (IR) File from the OpenVINO Integration with TensorFlow?

A Use the `openvino_tensorflow.export_ir(...)` API, as described here.

Q Do Developers Have to Update Application Code When Switching Between the OpenVINO Toolkit and the OpenVINO Integration with TensorFlow?

A Other than requiring two additional lines of code, the OpenVINO integration with TensorFlow doesn’t require any change in TensorFlow application code. Developers can continue to use their code with native TensorFlow APIs.

We want to give TensorFlow developers an easy way to adopt OpenVINO and become OpenVINO users. Once they see the value of OpenVINO optimization, we hope that they’ll graduate from an OpenVINO integration user to an OpenVINO toolkit developer, using our full suite of developer tools. That includes the Model Optimizer, Post Training Optimization Tool, and platforms such as Intel DevCloud. At that time, developers would need to start using the OpenVINO APIs to access the full functionality of the OpenVINO toolkit.
Installation Options and Dependencies

**Q** What Are We Releasing?

**A** We’re releasing the following binaries, git repos, and samples:

- Sample notebooks running on Intel DevCloud for the Edge: [Classification](#), [Object Detection](#)
- Sample Colab notebooks: [Classification example](#), [Object Detection example](#)
- Python wheel
  - `pip install openvino_tensorflow`
- GitHub repo
- Samples
- [Installation instructions](#) for supported OS/Python/TensorFlow versions
- Supported models

**Q** Where Can I Find all the Dependencies Packages Used to Build the Source?

**A** Please refer to the documentation. Note that we’re making many updates to the documentation.

Timeline and Future Developments

**Q** What’s the Timeline for the Next Release of the OpenVINO Integration with TensorFlow?

**A** We’re planning the next release in Q4. Beginning with OpenVINO 2022.1, the OpenVINO integration with TensorFlow will be completely in-sync with regular OpenVINO releases.

**Q** Are we Planning a Similar Integration with PyTorch or PaddlePaddle?

**A** For PaddlePaddle, contact Sesh R Seshagiri at sesh.r.seshagiri@intel.com.
  
  For PyTorch, we’ll likely have an update early next year.
Where Can I Go for Further Questions?

For product related issues, enhancement requests, or bugs, please use GitHub issues. We plan to use GitHub heavily for this project.

For general questions contact openvino-tensorflow@intel.com

* See backup for workloads and configurations. Results may vary.

1 Configurations: Intel® Core™ i7-9700K, 16GB memory, 300GB SSD, Ubuntu 18.04, TensorFlow version 2.4.1, OpenVINO™ integration with TensorFlow Version 0.5.0, Testing carried out by Extreme Vision. Full details of testing available here: https://bbs.cvmart.net/articles/5412


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 configuration disclosure for details. No product or component can be absolutely secure.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Intel technologies may require enabled hardware, software or service activation.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.01:22/3G/CAT/PDF ©Please Recycle 349616-001EN