APIs for Accelerating Vision and Inferencing
An Overview of Options and Trade-offs

Neil Trevett
Khronos President
NVIDIA VP Developer Ecosystems
ntrevett@nvidia.com | @neilt3d
Khronos is an open, non-profit, member-driven industry consortium developing royalty-free standards, and vibrant ecosystems, to harness the power of silicon acceleration for demanding graphics rendering and computationally intensive applications such as inferencing and vision processing.

>150 Members ~ 40% US, 30% Europe, 30% Asia

Some Khronos Standards Relevant to Embedded Vision and Inferencing

- OpenVX™: Vision and Inferencing Runtime
- NNEF™: Neural Network Exchange Format
- OpenCL™: Heterogeneous Parallel Compute
- SPIR™: Parallel Compute IR
- SYCL™: C++ Parallel Compute
- Vulkan®: GPU Acceleration

This work is licensed under a Creative Commons Attribution 4.0 International License.
Embedded Vision and Inferencing Acceleration

Networks trained on high-end desktop and cloud systems

Neural Network Training

Training Data

Ingestion

Compilation

Trained Networks

Compiled Code

Inferencing Engine

Application Code

Vision Library

Hardware Acceleration APIs

Sensor Data

Diverse Embedded Hardware (GPUs, DSPs, FPGAs)

Apps link to compiled code or inferencing library

Networks trained on high-end desktop and cloud systems

This work is licensed under a Creative Commons Attribution 4.0 International License
Neural Network Training Acceleration

Neural Network Training Frameworks

- PyTorch
- Caffe2
- TensorFlow
- Theano
- mxnet
- Chainer
- Caffe
- Keras
- PaddlePaddle

Neural Net Training Frameworks

Network training needs 32-bit floating-point precision

ONNX

Training Data

cuDNN

MIOpen

clDNN

Desktop and Cloud Hardware

Hardware Acceleration APIs

- NVIDIA CUDA
- AMD OpenCL
- Intel OpenCL
- SYCL

TPU
Embedded Vision and Inferencing Acceleration

Networks trained on high-end desktop and cloud systems

- Neural Network Training
- Trained Networks
- Training Data

Apps link to compiled code or inferencing library

- Compiled Code
- Inference Engine
- Application Code
- Vision Library

Diverse Embedded Hardware (GPUs, DSPs, FPGAs)

- Hardware Acceleration APIs
- Sensor Data
- GPU
- FPGA
- DSP
- Dedicated Hardware

This work is licensed under a Creative Commons Attribution 4.0 International License
Neural Network Exchange Formats

Before - Training and Inferencing Fragmentation

Training Framework 1 → Inference Engine 1
Training Framework 2 → Inference Engine 1
Training Framework 3 → Inference Engine 1

Every Inferencing Engine needs a custom importer from every Framework

After - NN Training and Inferencing Interoperability

Training Framework 1 → Neural Network Exchange Format
Training Framework 2 → Neural Network Exchange Format
Training Framework 3 → Neural Network Exchange Format

Common Optimization and processing tools

Two Neural Network Exchange Format Initiatives

<table>
<thead>
<tr>
<th>NNEF</th>
<th>ONNX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded Inferencing Import</td>
<td>Training Interchange</td>
</tr>
<tr>
<td>Defined Specification</td>
<td>Open Source Project</td>
</tr>
<tr>
<td>Multi-company Governance at Khronos</td>
<td>Initiated by Facebook &amp; Microsoft</td>
</tr>
<tr>
<td>Stability for hardware deployment</td>
<td>Software stack flexibility</td>
</tr>
</tbody>
</table>

ONNX and NNEF are Complementary
ONNX moves quickly to track authoring framework updates
NNEF provides a stable bridge from training into edge inferencing engines
NNEF and ONNX Industry Support

NNEF V1.0 released in August 2018
After positive industry feedback on Provisional Specification.
Maintenance update issued in September 2019
Extensions to V1.0 released for expanded functionality

ONNX 1.6 Released in September 2019
Introduced support for Quantization
ONNX Runtime being integrated with GPU inferencing engines such as NVIDIA TensorRT

NNEF Working Group Participants

ONNX Supporters
NNEF Tools Ecosystem

NNEF Model Zoo
Now available on GitHub. Useful for checking that ingested NNEF produces acceptable results on target system

NNEF adopts a rigorous approach to design lifecycle
Especially important for safety-critical or mission-critical applications in automotive, industrial and infrastructure markets

NNEF open source projects hosted on Khronos
NNEF GitHub repository under Apache 2.0
https://github.com/KhronosGroup/NNEF-Tools

Files

- TensorFlow and TensorFlow Lite Import/Export
- Caffe and Caffe2 Import/Export
- ONNX Import/Export
- Syntax Parser and Validator
- OpenVX Ingestion and Execution

Compound operations captured by exporting graph Python script
Embedded Vision and Inferencing Acceleration

Networks trained on high-end desktop and cloud systems

Apps link to compiled code or inferencing library

Diverse Embedded Hardware (GPUs, DSPs, FPGAs)

Neural Network Training

Training Data

Trained Networks

Compilation

Ingestion

Compiled Code

Inferencing Engine

Application Code

Vision Library

Hardware Acceleration APIs

Sensor Data

This work is licensed under a Creative Commons Attribution 4.0 International License
# Primary Machine Learning Compilers

<table>
<thead>
<tr>
<th>Import Formats</th>
<th>Caffe, Keras, MXNet, ONNX</th>
<th>TensorFlow Graph, MXNet, Paddle, Keras, ONNX</th>
<th>PyTorch, ONNX</th>
<th>TensorFlow Graph, PyTorch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front-end / IR</td>
<td>NNVM / Relay IR</td>
<td>nGraph / Stripe IR</td>
<td>Glow Core / Glow IR</td>
<td>XLA HLO</td>
</tr>
<tr>
<td>Output</td>
<td>OpenCL, LLVM, CUDA, Metal</td>
<td>OpenCL, LLVM, CUDA</td>
<td>OpenCL LLVM</td>
<td>LLVM, TPU IR, XLA IR TensorFlow Lite / NNAPI (inc. HW accel)</td>
</tr>
</tbody>
</table>

![Diagram of machine learning compilers](image)
ML Compiler Steps

Embedded NN Compilers
- CEVA Deep Neural Network (CDNN)
- Cadence Xtensa Neural Network Compiler (XNNC)

Consistent Steps
1. Import Trained Network Description
2. Apply graph-level optimizations e.g. node fusion, node lowering and memory tiling
3. Decompose to primitive instructions and emit programs for accelerated run-times

Fast progress but still area of intense research
If compiler optimizations are effective - hardware accelerator APIs can stay ‘simple’ and won’t need complex metacommands (combined primitive commands) like DirectML

Google’s Multi-Level IR
Enables multiple domain-specific dialects within a common framework
Experimenting with emitting to Vulkan/SPIR-V

<table>
<thead>
<tr>
<th>Import Formats</th>
<th>Front-end / IR</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffe, Keras, MXNet, ONNX</td>
<td>NNVM / Relay IR</td>
<td>OpenCL, LLVM, CUDA, Metal</td>
</tr>
<tr>
<td>TensorFlow Graph, MXNet, PaddlePaddle, Keras, ONNX</td>
<td>nGraph / Stripe IR</td>
<td>OpenCL, LLVM, CUDA</td>
</tr>
<tr>
<td>PyTorch, ONNX</td>
<td>Glow Core / Glow IR</td>
<td>OpenCL, LLVM</td>
</tr>
<tr>
<td>TensorFlow Graph, PyTorch</td>
<td>XLA HLO</td>
<td>LLVM, TPU IR, XLA IR, TensorFlow Lite / NNAPI (inc. HW accel)</td>
</tr>
</tbody>
</table>
SPIR-V Ecosystem

Third party kernel and shader languages

GLSL
HLSL

Third party kernel and shader languages

glslang
DXC

SPIR-V (Dis)Assembler
SPIRV-Cross
SPIRV-opt | SPIRV-remap

Optimization Tools

Environment spec for each target API used to drive compilation

GLSL
HLSL

OpenCL C
Front-end
OpenCL C++
Front-end

SYCL for OpenCL
Front-end

C++ for ISO C++

SPIR-V
Khronos-defined cross-API IR
Native graphics and parallel compute support
Easily parsed/extended 32-bit stream
Data object/control flow retained for effective code generation/translation

SPIR-V

https://github.com/KhronosGroup/SPIRV-Tools

KHRONOS GROUP

This work is licensed under a Creative Commons Attribution 4.0 International License
Embedded Vision and Inferencing Acceleration

Networks trained on high-end desktop and cloud systems

Apps link to compiled code or inferencing library

Diverse Embedded Hardware (GPUs, DSPs, FPGAs)

Neural Network Training

Training Data

Trained Networks

Compilation

Ingestion

Compiled Code

Inferencing Engine

Application Code

Vision Library

Hardware Acceleration APIs

Sensor Data

Compiled Code

Inferencing Engine

Application Code

Vision Library

Diverse Embedded Hardware (GPUs, DSPs, FPGAs)
PC/Mobile Platform Inferencing Stacks

Consistent Steps
1. Import and optimize trained NN model file
2. Ingest graph into ML runtime
3. Accelerate inferencing operations using underlying low-level API

Microsoft Windows
Windows Machine Learning (WinML)

Google Android
Neural Network API (NNAPI)
https://developer.android.com/ndk/guides/neuralnetworks/

Apple MacOS and iOS
CoreML
https://developer.apple.com/documentation/coreml

ONNX
Application #1
Input Surface
Application #2
Output Surface
WinML RT API
WinML Win32 API
WinML Runtime
Model Inference Engine
DirectML
Direct3D
CPU
GPU

TensorFlow Lite
Application
Machine learning framework/library
Android Neural Networks Runtime
Hardware acceleration
CPU fallback

Android NN API
Vendor NN DSP driver
Vendor NN driver
Vendor NN GPU driver

Apple MacOS and iOS
CoreML

Core ML Model
Vision
Natural language processing
GameplayKit
Core ML
Accelerate and BNNS
Metal Performance Shaders
Inferencing Engines

Platform Engines
- Android NNAPI
- Microsoft WinML
- Apple CoreML

Desktop IHVs
- AMD MIVisionX over MIOpen
- Intel OpenVINO over cLDNN
- NVIDIA TensorRT over cuDNN

Mobile / Embedded
- Arm Compute Library
- Huawei MACE
- Qualcomm Neural Processing SDK
- Synopsis MetaWare EV Dev Toolkit
- TI Deep Learning Library (TIDL)
- VeriSilicon Acuity

Acceleration APIs
- Vulkan
- OpenGL ES
- DirectX 12
- OpenCL
- CUDA

Cross-platform Inferencing Engines
- OpenCV
- OpenVX

Almost all Embedded Inferencing Engines use OpenCL to access accelerator silicon

Both provide Inferencing AND Vision acceleration
### OpenVX and OpenCV are Complementary

<table>
<thead>
<tr>
<th></th>
<th>OpenCV</th>
<th>OpenVX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Implementation</strong></td>
<td>Community driven open source library</td>
<td>Callable API implemented, optimized and shipped by hardware vendors</td>
</tr>
<tr>
<td><strong>Scope</strong></td>
<td>100s of imaging and vision functions</td>
<td>Tight focus on dozens of core hardware accelerated functions plus extensions and accelerated custom nodes. Uses external camera drivers</td>
</tr>
<tr>
<td></td>
<td>Multiple camera APIs/interfaces</td>
<td></td>
</tr>
<tr>
<td><strong>Conformance</strong></td>
<td>Extensive OpenCV Test Suite but no formal Adopters program</td>
<td>Implementations must pass Khronos Conformance Test Suite to use trademark</td>
</tr>
<tr>
<td><strong>IP Protection</strong></td>
<td>None. Source code licensed under BSD. Some modules require royalties/licensing</td>
<td>Protected under Khronos IP Framework - Khronos members agree not to assert patents against API when used in Conformant implementations</td>
</tr>
<tr>
<td><strong>Acceleration</strong></td>
<td>OpenCV 3.0 Transparent API (or T-API) enables function offload to OpenCL devices</td>
<td>Implementation free to use any underlying API such as OpenCL. Can use OpenCL for Custom Nodes</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>OpenCV 4.0 G-API graph model for some filters, arithmetic(binary) operations, and well-defined geometrical transformations</td>
<td>Graph-based execution of all Nodes. Optimizable computation and data transfer</td>
</tr>
<tr>
<td><strong>Inferencing</strong></td>
<td>Deep Neural Network module to construct networks from layers for forward pass computations only. Import from ONNX, TensorFlow, Torch, Caffe</td>
<td>Neural Network Layers and operations represented directly in the OpenVX Graph. NNEF direct import, ONNX through NNEF convertor</td>
</tr>
</tbody>
</table>
OpenVX Cross-Vendor Inferencing

OpenVX
A high-level graph-based abstraction for portable, efficient vision processing
Optimized OpenVX drivers created and shipped by processor vendors
Can be implemented on almost any hardware or processor
Graph can contain vision processing and NN nodes - enables global optimizations
Run-time graph execution can be almost completely autonomously from the host CPU

Performance comparable to hand-optimized, non-portable code
Real, complex applications on real, complex hardware
Much lower development effort than hand-optimized code
Extending OpenVX with Custom Nodes

**OpenVX/OpenCL Interop**
- Provisional Extension
- Enables custom OpenCL acceleration to be invoked from OpenVX User Kernels
- Memory objects can be mapped or copied

**Kernel/Graph Import**
- Provisional Extension
- Defines container for executable or IR code
- Enables arbitrary code to be inserted as a OpenVX Node in a graph

OpenVX user-kernels can access command queue and cl_mem objects to asynchronously schedule OpenCL kernel execution.

Application

OpenVX data objects

OpenCL Command Queue

Copy or export cl_mem buffers into OpenVX data objects

Map or copy OpenVX data objects into cl_mem buffers

cl_mem buffers

Runtime

Fully asynchronous host-device operations during data exchange

Runtime
OpenVX 1.3 Announced This Week!

OpenVX 1.3 Feature Sets
Enables deployment flexibility while avoiding fragmentation
Implementations with one or more complete feature sets are conformant
- Baseline Graph Infrastructure (enables other Feature Sets)
- Default Vision Functions
- Enhanced Vision Functions (introduced in OpenVX 1.2)
- Neural Network Inferencing (including tensor objects)
  - NNEF Kernel import (including tensor objects)
  - Binary Images
- Safety Critical (reduced features for easier safety certification)

Open Source Prototype OpenVX 1.3
Conformance Test Suite
Finalization expected before the end of 2019
https://github.com/KhronosGroup/OpenVX-cts/tree/openvx_1.3

Open Source OpenVX Tutorial
and Code Samples
https://github.com/rgiduthuri/openvx_tutorial

Open source OpenVX 1.3 for Raspberry Pi
Raspberry Pi 3 Model B with Raspbian OS
Automatic optimization of memory access patterns via tiling and chaining
Highly optimized kernels leveraging multimedia instruction set
Automatic parallelization for multicore CPUs and GPUs
Automatic merging of common kernel sequences
https://github.com/KhronosGroup/OpenVX-sample-impl/tree/openvx_1.3

© The Khronos® Group Inc. 2019 - Page 19
This work is licensed under a Creative Commons Attribution 4.0 International License
**GPU and Heterogeneous Acceleration APIs**

- **GPU APIs**
  - Application
  - GPU

- **Heterogenous APIs**
  - Application
  - OpenCL
  - Heterogeneous Compute Resources
  - CPU
  - GPU
  - FPGA
  - DSP
  - Custom Hardware

Growing number of optimized OpenCL libraries
- Vision and Imaging
- Machine Learning and Inferencing
- Linear Algebra and Mathematics
- Physics

OpenCL provides a programming and runtime framework
- OpenCL C and C++ kernels can be compiled and loaded to any available device
- Low-level control over memory allocation and parallel task execution
- Simpler, relatively lightweight and more flexible than GPU APIs

- Only Vulkan is cross platform API with rendering - including Embedded Linux and Android
- GPU Rendering APIs gradually expanding compute capabilities e.g. short data types
- Many mobile SOCs and Embedded Systems becoming increasingly heterogeneous
  - Autonomous vehicles
  - Vision and inferencing

- Only Vulkan is cross platform API with rendering - including Embedded Linux and Android
- GPU Rendering APIs gradually expanding compute capabilities e.g. short data types
- Many mobile SOCs and Embedded Systems becoming increasingly heterogeneous
  - Autonomous vehicles
  - Vision and inferencing
OpenCL is Widely Deployed and Used

Hardware Implementations
- Apple
- ATERA
- AMD
- arm
- IBM
- Imagination
- Intel
- MARVELL
- MediaTek
- Qualcomm
- NVIDIA
- Samsung
- ST
- Texas Instruments
- VeriSilicon
- Xilinx

Desktop Creative Apps
- Adobe
- blender
- Capture One
- CyberLink
- CHAOGROUP
- ptc
- Blackmagic Design
- GIMP
- Autodesk

Linear Algebra Libraries
- CLBlast
- ViennaCL
- SYCL-BLAS

Parallel Computation Languages
- OpenACC
- SYCL
- aparapi
- PyOpenCL

Math and Physics Libraries
- ArrayFire
- MATHLAB
- Wolfram Mathematica
- C++ Libraries
- GNU Octave

Vision and Imaging Libraries
- VisionCpp
- Halide
- OpenVX
- OpenCV

Machine Learning Inferencing Compilers
- Halide
- OpenVX
- OpenCL
- Arm Compute Library

Machine Learning Libraries
- Android NNAPI
- Caffe
- TensorFlow
- Qualcomm Neural Processing SDK for AI
- SYCL-DNN
- TI Deep Learning Library (TIDL)
- OpenVINO
- Intel MACE
- Intel cLIDNN
- Synopsis MetaWare EV
- VeriSilicon
- Acuity
SYCL Single Source C++ Parallel Programming

- SYCL 1.2.1 Adopters Program released in July 2018

- Multiple SYCL libraries for vision and inferencing
  - SYCL-BLAS, SYCL-DNN, SYCL-Eigen, SYCL Parallel STL

C++ Kernel Fusion can give better performance on complex apps and libs than hand-coding

Ideal for accelerating larger C++-based engines and applications

Accelerated code passed into device OpenCL compilers

E.g. complex ML frameworks can be directly compiled and accelerated

This work is licensed under a Creative Commons Attribution 4.0 International License
SYCL Implementations

This work is licensed under a Creative Commons Attribution 4.0 International License

SYCL enables Khronos to influence ISO to enable standard C++ to (eventually) support heterogenous compute

Multiple Backend Support Coming
SYCL beginning to be supported on low-level APIs in addition to OpenCL e.g. Vulkan and CUDA
http://sycl.tech

Intel Adoption
Intel’s ‘One API’ Initiative uses SYCL

LLVM/clang SYCL Compiler
Compiles C++-based SYCL source files into code for both CPU and a wide range of compute accelerators

ComputeCpp
Codeplay Software’s v1.2.1 conformant implementation available to download today

triSYCL
Open-source test-bed to experiment with the specification of the OpenCL SYCL C++ layer and to give feedback to Khronos

HipSYCL
SYCL 1.2.1 implementation that builds upon NVIDIA CUDA/AMD HIP/ROCM
OpenCL Evolution

OpenCL Extension Specs
- Scratch-Pad Memory Management
- Vulkan / OpenCL Interop
- Extended Subgroups
- SPIR-V 1.4 ingestion for compiler efficiency
- SPIR-V Extended debug info

Regular Maintenance Updates
- Regular updates for spec clarifications, formatting and bug fixes
- https://www.khronos.org/registry/OpenCL/

Integration of Extensions plus New Core functionality

Focus for OpenCL Next is ‘Deployment Flexibility’
- Flexible Profile enables embedded vendors to ship targeted functionality for their customers and be officially conformant

Repeat Cycle for next Core Specification
- Target 2020 ‘OpenCL Next’
- May 2017 OpenCL 2.2
Pervasive Vulkan

Major GPU Companies supporting Vulkan for Desktop and Mobile Platforms

Platforms

Desktop

Android (Android 7.0+)
(Vulkan 1.1 required on Android Q)

Apple (via porting layers)

Media Players

Consoles

Virtual Reality

Cloud Services

Game Streaming

Embedded

Game Engines

EPIC GAMES
id
CRYENGINE
unity
VALVE
Croteam
Serious Engine
XENKO

© The Khronos® Group Inc. 2019 - Page 26

This work is licensed under a Creative Commons Attribution 4.0 International License
Safety Critical GPU API Evolution

OpenGL ES 1.0 - 2003
Fixed function graphics

OpenGL ES 2.0 - 2007
Programmable Shaders

OpenGL SC 1.0 - 2005
Fixed function graphics safety critical subset

OpenGL SC 2.0 - April 2016
Programmable Shaders safety critical subset

Vulkan 1.0 - 2016
Explicit Graphics and Compute

Potential OpenCL SC work will leverage the deployment flexibility of ‘OpenCL Next’ to minimize API surface area

Vulkan is Compelling Starting Point for SC GPU API Design
• Widely adopted, royalty-free open standard
• Low-level explicit API - smaller surface area than OpenGL
• Not burdened by debug functionality
  • Very little internal state
  • Well-defined thread behavior

ISO 26262

Clearlly Definable Design Goals to Adapt Vulkan for SC
• Reduce driver size and complexity
  • Offline pipeline creation, no dynamic display resolutions
• Deterministic Behavior
  • No ignored parameters, static memory management, eliminate undefined behaviors
• Robust Error Handling
  • Error callbacks so app can respond, Fatal error callbacks for fast recovery initiation

Khronos Vulkan SC Working Group started work in February 2019

This work is licensed under a Creative Commons Attribution 4.0 International License

© The Khronos® Group Inc. 2019 - Page 27
Need for New Camera Control API Standard?

- Khronos suspended work on OpenKCam standard several years ago
  - Mobile market went proprietary - but embedded market has different needs

- OpenKCAM was aiming at advanced control of ISP and camera with cross-platform portability
  - Generate sophisticated image stream for advanced imaging & vision apps
  - Portable access to growing sensor diversity: e.g. depth sensors and sensor arrays
  - Cross sensor synch: e.g. synch of multiple camera and MEMS sensors
  - Advanced, high-frequency per-frame burst control of camera/sensor: e.g. ROI
  - Multiple input, output re-circulating streams with RAW, Bayer or YUV Processing

Liaison opportunity for cooperation - maybe over OpenKCam and GeniCam?
Work together to integrate next generation camera control and acceleration APIs?

Image Signal Processor (ISP)

Defines control of Sensor, Color Filter Array
Lens, Flash, Focus, Aperture

Auto Exposure (AE)
Auto White Balance (AWB)
Auto Focus (AF)

Stream of images for downstream processing
Thank You and Resources

- **Khronos Standards:** OpenVX, NNEF, OpenCL, SPIR, SYCL, Vulkan and more...
  - Any company is welcome to join Khronos [www.khronos.org](http://www.khronos.org)
  - OpenVX 1.3: [www.khronos.org/openvx/](http://www.khronos.org/openvx/)
  - SYCL: [http://sycl.tech](http://sycl.tech)

- **Compilers for Machine Learning Conference Proceedings:** [www.c4ml.org](http://www.c4ml.org)

- **MLIR:** [www.blog.google/technology/ai/mlir-accelerating-ai-open-source-infrastructure/](http://www.blog.google/technology/ai/mlir-accelerating-ai-open-source-infrastructure/)

- **Neil Trevett:** ntrevett@nvidia.com | @neilt3d

---

**Benefits of Khronos membership**

- Gain early insights into industry trends and directions
- Influence the design and direction of key open standards that will drive your business
- Accelerate your time-to-market with early access to specification drafts
- Network with domain experts from diverse companies in your industry
- Draft Specifications Confidential to Khronos members
- State-of-the-art IP Framework protects your Intellectual Property
- Publicly Release Specifications and Conformance Tests
- Enhance your company reputation as an industry leader through Khronos participation

This work is licensed under a Creative Commons Attribution 4.0 International License

© The Khronos® Group Inc. 2019 - Page 29