High performance applications using OpenVINO™

Yury Gorbachev
Why?

• DL Training and inference are essentially very different domains
  • Frameworks contain too much for training
  • Hardware requirements are very different (100W is pretty normal)
  • Performance goals are different (batch size, latency vs. throughput)
  • Environment is different (development vs. deployment)

• Hard to find solution that does DL inference properly
What is proper Deep Learning inference?

- Highest performance characteristics
  - Inference/watt/$ is major concern, best possible performance is a must
- Minimal footprint
  - Memory, binary size, execution overhead
- Absolute minimum of dependencies
- Cross platform portability
- Backward compatibility and predictable maintenance
Also

• CV application is not just Deep Learning
• “Classical” components widely used

• OpenCV is undisputable champion of the CV world
• Need to satisfy deployment requirements as well
  • Performance, footprint, legal cleanness, etc.
OpenVINO™ Toolkit for best CV/DL applications

- Development toolkit for high performance CV and DL inference
  - Solution for application designers
  - No training/research overhead or specifics, minimal footprint, highly portable code
- Set of libraries to solve CV/DL deployment problems
  - Fastest OpenCV build
  - Deep Learning Inference Engine
- Provides access to all Intel accelerators and heterogeneous execution model
  - Intel CPU, integrated GPU
  - Vision Processing Unit (VPU) and FPGA
OpenVINO™ vs. Computer Vision

Computer Vision Pipeline

Input -> Custom Components -> CV/non-DL Components -> DL Components

Custom Components

CV/non-DL Components

DL Components

Object Person Face Emotion Gesture Text ...

Custom Code

Direct Coding Solution

API Solution

Caffe

TensorFlow

OpenVINO™

API Solution

DL Inference Engine

OpenCV

OpenVX

VPU GPU FPGA CPU VPU GPU FPGA CPU VPU GPU FPGA CPU
OpenCV: OpenVINO vs. Open Source

- Most performant and fine tuned build
  - SSE, AVX2 vectorization & TBB/OMP parallelism
  - GPU offload via Transparent API
  - DL Inference Engine by default for OpenCV DNN
- Legally clean
  - Each build is checked with IP Protection tools, safe for production
- Additional algorithms from Intel in binary form
  - Face Detection and analysis libraries
Deep Learning Inference Engine

- Pure inference oriented solution (no training included)
  - Superior performance on Intel platform, highly optimized
  - Minimal memory use
- No framework required in runtime
- Support for CPU, GPU, FPGA, Movidius
  - Heterogeneous execution support
- Cross-platform portability
Deep Learning performance using OpenVINO/CPU

![Bar chart showing performance comparison between different models and platforms]

- **DenseNet-121**:
  - Caffe/MKL CPU fp32: 5.4
  - OpenVINO IE CPU fp32: 3.6

- **ResNet-50**:
  - Caffe/MKL CPU fp32: 3.5
  - OpenVINO IE CPU fp32: 3.0

- **GoogleNet V4**:
  - Caffe/MKL CPU fp32: 2.7

- **SSD 300**:
  - Caffe/MKL CPU fp32: 3.5

- **Yolo V2**:
  - Caffe/MKL CPU fp32: 2.7

**Core™-i5 6500@2.9 GHZ**
Deep Learning performance using OpenVINO/GPU

Core™ i5 6500@2.9 GHz
Inference memory footprint on CPU

- DenseNet-121: 7.8 MB
- GoogleNet V4: 3.0 MB
- ResNet-50: 3.1 MB
- SSD 300: 3.4 MB
- MobileNet SSD: 4.5 MB
- Yolo V2: 1.6 MB

Core™-i5 6500@2.9 GHZ
With Deep Learning Frameworks

Pre-trained Models → Model → User Application + Framework
Video/Image

With OpenVINO™ DL Inference Engine

Pre-trained Models → Model → Model Optimizer → Once in Design Time → IR → User Application + Inference Engine
Video/Image
Deep Learning Inference Engine (IE)

1. Single API solution across accelerators
2. Heterogeneous network execution across accelerators
3. Framework independent lightweight internal representation
4. Customizations in C++ and OpenCL languages

Deep Learning application

1. Design time
2. Model Optimizer
3. IR
4. C++ layers
   - MKL-DNN
   - clDNN
   - Custom OCL layers
   - MVNC
   - DLA

DL Inference Engine API

Heterogeneous Execution Engine

- CPU Plugin
- GPU Plugin
- Movidius Plugin
- FPGA Plugin

Custom layers
Easier deployment

- Accurate against original framework
  - Direct replacement of original framework calls
  - No retraining/fine-tuning required
- Unified support for multiple OSes
  - Linux and Windows are equally supported and performant
- Encapsulates basic preprocessing
  - Mean subtraction / normalization integrated into model
Portability across platforms

• Single API across platforms
  • No need to change SDKs and application codes depending on targets
• Consistent set of layers and accurate results across targets
  • Verified against reference model/framework
• Heterogeneous execution for missing pieces
  • CPU fallback whenever needed
Additional portability benefits

• Intel has rapidly developing hardware set
  • New SoCs and architectures are evolving
• No need to wait for HW itself or emulator
  • Design app for existing targets first and move to new ones easy
• (Bonus) Check algos on fastest, deploy on most suitable!

Create application using Inference Engine API
Design and validate on CPU/GPU clusters
Deploy on Movidius/FPGA Targets
Customization possibilities

- OpenVINO™ is partially binary product right now
- Possible to implement and add own layers
  - New topologies are easy to support
- Most of the layers are delivered as a source
  - Check how it is done for known topologies
  - Base your implementation on those for faster TTM
- MKLDNN for CPU and cLDNN for GPU are already available in source form
  - Possible to check implementation there
Open Model Zoo

• Free reference models for Deep Learning Inference Engine
  • Object Detection (Face, People, Vehicles, etc.)
  • Object Analysis (Facial analysis, Head Pose, Vehicle attributes)
• Superior performance on Intel
  • Core™ i5 CPU: SSD 300 (6 fps) vs. People Detection Model (68 fps)

Significant reduction in development efforts, no dataset & training needed
Samples

• Basic samples to facilitate API understanding
  • Classification, object detection, segmentation
  • Target selection via command line

• Extended samples using Model Zoo
  • Face analysis, Security camera sample

• Interworking between Media SDK, OpenCV, DL IE
  • Automated public models downloader script