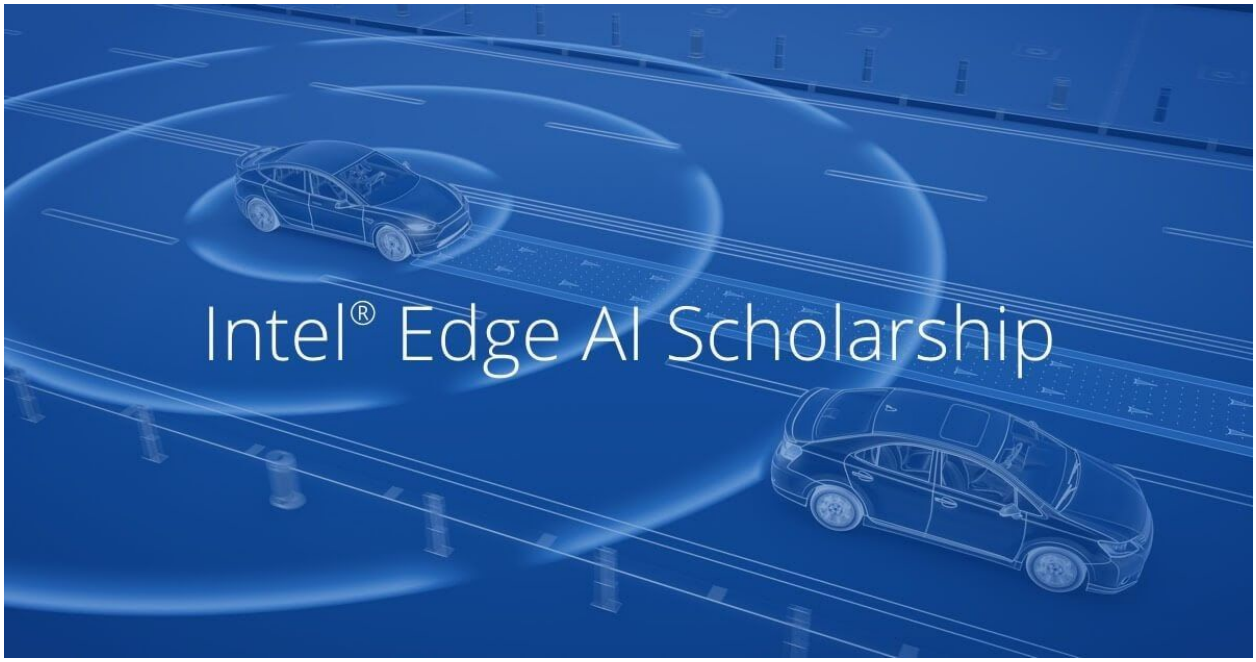

Frequently Used Resources For Foundation Course

This document contains the most recommended resources for all the prerequisites and topics related to the Intel Edge AI Foundation Course.

Intel OpenVINO Toolkit

1. Intel Distribution of OpenVINO toolkit Documentation
   https://software.intel.com/en-us/openvino-toolkit

2. Introduction to OpenVINO toolkit
   https://towardsdatascience.com/introduction-to-openvino-897e705a1f0a

3. Introduction to Intel Distribution of OpenVINO toolkit for Computer Vision Applications Coursera
   https://www.coursera.org/learn/intel-openvino

4. How does the Intel® Distribution of OpenVINO™ toolkit differ from the open-source version?
   https://01.org/openvinooolkit/faq

Python

1. Introduction to Python Programming Udacity
   https://www.udacity.com/course/introduction-to-python--ud1110

2. A complete introduction to Python
   https://medium.com/free-code-camp/learning-python-from-zero-to-hero-120ea540b567

3. Python Project: pillow, tesseract, and OpenCV
   https://www.coursera.org/learn/python-project

4. Intro to Data Structures and Algorithms with Python
   https://www.udacity.com/course/data-structures-and-algorithms-in-python--ud513

OpenCV

1. OpenCV with Python by Example (A very helpful Packtpub book written by Prateek Joshi)

2. A guide to learning OpenCV
   https://www.pyimagesearch.com/2018/07/19/opencv-tutorial-a-guide-to-learn-opencv/

Machine Learning

1. Introduction to Machine Learning Udacity
   https://www.udacity.com/course/intro-to-machine-learning--ud120

2. Machine Learning offered by Stanford University
   https://www.coursera.org/learn/machine-learning

3. Machine Learning Interview Preparation
   https://www.udacity.com/course/machine-learning-interview-prep--ud1001

Deep Learning

1. Deep Learning Specialization by deeplearning.ai
   https://www.coursera.org/specializations/deep-learning

2. Introduction to Deep Learning in PyTorch Udacity
   https://www.udacity.com/course/deep-learning-pytorch--ud188

3. Applied Deep Learning with TensorFlow offered by Intel AI Academy
   https://software.intel.com/en-us/ai/courses/tensorflow

TensorFlow

1. TensorFlow in Practice Specialization by deeplearning.ai
   https://www.coursera.org/specializations/tensorflow-in-practice

2. Introduction to TensorFlow for Deep Learning Udacity
   https://www.udacity.com/course/intro-to-tensorflow-for-deep-learning--ud187

Computer Vision

1. Introduction to Computer Vision Udacity
   https://www.udacity.com/course/introduction-to-computer-vision--ud810

2. Introduction to Computer Vision with Watson and OpenCV Coursera
   https://www.coursera.org/learn/introduction-computer-vision-watson-opencv

3. Self-driving Cars Specialization Coursera
   https://www.coursera.org/specializations/self-driving-cars

Edge Computing

1. What is Edge Computing?
   https://medium.com/@miccowang/what-is-edge-computing-f997c0ab39fc

2. From Cloud Computing to Edge Computing
   https://medium.com/@opensourcevoices/from-cloud-computing-to-edge-computing-63ad1f624f14

AI at the Edge

1. AI on the Edge with Computer Vision
   https://software.intel.com/en-us/ai/courses/ai-on-the-edge-computer-vision

2. AI from the Data Center to the Edge – An Optimized Path using Intel Architecture
   https://software.seek.intel.com/DataCenter_to_Edge_REG

3. A good article on AI at the Edge
   https://medium.com/@Edgify/ai-at-the-edge-as-a-way-to-redefine-the-meaning-of-privacy-52ef647fbb52

Pre-Trained Models

1. List of all the available Pre-Trained models in the Intel Distribution of OpenVINO Toolkit
   https://software.intel.com/en-us/openvino-toolkit/documentation/pretrained-models

2. OpenVINO Toolkit Documentation for Pre-Trained Models
   https://docs.openvinotoolkit.org/2019_R1/_docs_Pre_Trained_Models.html

Model Optimizer

1. Model Optimizer Developer Guide
   https://docs.openvinotoolkit.org/latest/_docs_MO_DG_Deep_Learning_Model_Optimizer_DevGuide.html

2. Frequently Asked Questions related to Model Optimizer
   https://docs.openvinotoolkit.org/latest/_docs_MO_DG_prepare_model_Model_Optimizer_FAQ.html

3. Intel Article about using Model Optimizer and Inference Engine for Intel Distribution of OpenVINO Toolkit on Raspberry Pi
   https://software.intel.com/en-us/articles/model-downloader-optimizer-for-openvino-on-raspberry-pi

Inference Engine

1. Inference Engine Developer Guide
   https://docs.openvinotoolkit.org/latest/_docs_IE_DG_Deep_Learning_Inference_Engine_DevGuide.html

Deploying Edge Applications Using OpenVINO

1. OpenVINO Toolkit Code Samples and Demos
   https://software.intel.com/en-us/openvino-toolkit/documentation/code-samples

2. Object Recognition with Intel Distribution of OpenVINO Toolkit
   https://medium.com/intel-software-innovators/object-recognition-with-intel-distribution-of-openvino-toolkit-475647574fb7

3. Smart Bird Watcher using Intel Distribution of OpenVINO Toolkit
   https://towardsdatascience.com/smart-bird-watcher-customizing-pre-trained-ai-models-to-detect-birds-of-interest-dca1202bfbdf

Articles written by "Udacity Intel Edge AI Scholarship" Students
https://medium.com/udacity-intel-edge-ai-scholars-2k19-20

Links for further Research in the Foundation Course

- Edge Inference with Intel® DevCloud
- Image Classification vs. Object Detection vs. Image Segmentation
- SSD: Single Shot MultiBox Detector
- You Only Look Once (YOLO): Unified, Real-Time Object Detection
- Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
- MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications
- Deep Residual Learning for Image Recognition (ResNet)
- Inception Research Paper
- Understanding SSD MultiBox — Real-Time Object Detection In Deep Learning
- Going beyond the bounding box with semantic segmentation
- Quantization - Neural Network Distiller
- Model Optimization Techniques (OpenVINO Documentation)
- Converting a Model to Intermediate Representation (IR)
- Model Optimizer Supported Framework Layers
- Intermediate Representation Notation Reference Catalog

- TensorFlow Model Zoo
- Cutting Off Parts of a Model
- Custom Layers in the Model Optimizer
- Offloading Sub-Graph Inference to TensorFlow
- Supported Devices
- IECore Class Reference
- Inference Engine Python API Reference
- IENetwork Class Reference
- ExecutableNetwork Class Reference
- InferRequest Class Reference
- Synchronous Vs Asynchronous
- Intel Internet of Things Applications Across Industries
- The Ultimate Guide To Starting Your First IoT Project
- OpenVINO, OpenCV, and Movidius NCS on the Raspberry Pi
- What is the best programming language for Machine Learning?
- Optimization Guide
- OpenCV Tutorials
- MQTT.org
- What is MQTT and How Does it Work?

Lesson Glossaries

Lesson 2

Edge Application

Applications with inference run on local hardware, sometimes without network connections, such as Internet of Things (IoT) devices, as opposed to the cloud. Less data needs to be streamed over a network connection, and real-time decisions can be made.

OpenVINO™ Toolkit

The Intel® Distribution of OpenVINO™ Toolkit enables deep learning inference at the edge by including both neural network optimizations for inference as well as hardware-based optimizations for Intel® hardware.

Pre-Trained Model

Computer Vision and/or AI models that are already trained on large datasets and available for use in your own applications. These models are often trained on datasets like ImageNet. Pre-trained models can either be used as-is or used in transfer learning to further fine-tune a model. The OpenVINO™ Toolkit provides a number of pre-trained models that are already optimized for inference.

Transfer Learning

The use of a pre-trained model as a basis for further training of a neural network. Using a pre-trained model can help speed up training as the early layers of the network have feature extractors that work in a wide variety of applications, and often only late layers will need further fine-tuning for your own dataset. OpenVINO™ does not deal with transfer learning, as all training should occur prior to using the Model Optimizer.

Image Classification

A form of inference in which an object in an image is determined to be of a particular class, such as a cat vs. a dog.

Object Detection

A form of inference in which objects within an image are detected and a bounding box is output based on where in the image the object was detected. Usually, this is combined with some form of classification to also output which class the detected object belongs to.

Semantic Segmentation

A form of inference in which objects within an image are detected and classified on a pixel-by-pixel basis, with all objects of a given class given the same label.

Instance Segmentation

Similar to semantic segmentation, this form of inference is done on a pixel-by-pixel basis, but different objects of the same class are separately identified.

SSD

A neural network combining object detection and classification, with different feature extraction layers directly feeding to the detection layer, using default bounding box sizes and shapes/

YOLO

One of the original neural networks to only take a single look at an input image, whereas earlier networks ran a classifier multiple times across a single image at different locations and scales.

Faster R-CNN

A network, expanding on R-CNN and Fast R-CNN, that integrates advances made in the earlier models by adding a Region Proposal Network on top of the Fast R-CNN model for an integrated object detection model.

MobileNet

A neural network architecture optimized for speed and size with minimal loss of inference accuracy through the use of techniques like 1x1 convolutions. As such, MobileNet is more useful in mobile applications that substantially larger and slower networks.

ResNet

A very deep neural network that made use of residual, or "skip" layers that pass information forward by a couple of layers. This helped deal with the vanishing gradient problem experienced by deeper neural networks.

Inception

A neural network making use of multiple different convolutions at each "layer" of the network, such as 1x1, 3x3 and 5x5 convolutions. The top architecture from the original paper is also known as GoogLeNet, an homage to LeNet, an early neural network used for character recognition.

Inference Precision

Precision refers to the level of detail to weights and biases in a neural network, whether in floating-point precision or integer precision. Lower precision leads to lower accuracy, but with a positive trade-off for network speed and size.

Lesson 3

Model Optimizer

A command-line tool used for converting a model from one of the supported frameworks to an Intermediate Representation (IR), including certain performance optimizations, that is compatible with the Inference Engine.

Optimization Techniques

Optimization techniques adjust the original trained model in order to either reduce the size of or increase the speed of a model in performing inference. Techniques discussed in the lesson include quantization, freezing and fusion.

Quantization

Reduces precision of weights and biases (to lower precision floating-point values or integers), thereby reducing compute time and size with some (often minimal) loss of accuracy.

Freezing

In TensorFlow, this removes metadata only needed for training, as well as converting variables to constants. Also a term in training neural networks, where it often refers to freezing layers themselves in order to fine-tune only a subset of layers.

Fusion

The process of combining certain operations together into one operation and thereby needing less computational overhead. For example, a batch normalization layer, activation layer, and convolutional layer could be combined into a single operation. This can be particularly useful for GPU inference, where the separate operations may occur on separate GPU kernels, while a fused operation occurs on one kernel, thereby incurring less overhead in switching from one kernel to the next.

Supported Frameworks

The Intel® Distribution of OpenVINO™ Toolkit currently supports models from five frameworks (which themselves may support additional model frameworks): Caffe, TensorFlow, MXNet, ONNX, and Kaldi.

Caffe

The “Convolutional Architecture for Fast Feature Embedding” (CAFFE) framework is an open-source deep-learning library originally built at UC Berkeley.

TensorFlow

TensorFlow is an open-source deep-learning library originally built at Google. As an Easter egg for anyone who has read this far into the glossary, this was also your instructor’s first deep learning framework they learned, back in 2016 (pre-V1!).

MXNet

Apache MXNet is an open-source deep-learning library built by Apache Software Foundation.

ONNX

The “Open Neural Network Exchange” (ONNX) framework is an open-source deep-learning library originally built by Facebook and Microsoft. PyTorch and Apple-ML models are able to be converted to ONNX models.

Kaldi

While still open-source like the other supported frameworks, Kaldi is mostly focused around speech recognition data, with the others being more generalized frameworks.

Intermediate Representation

A set of files converted from one of the supported frameworks, or available as one of the Pre-Trained Models. This has been optimized for inference through the Inference Engine, and maybe at one of several different precision levels. Made of two files:

.xml - Describes the network topology

.bin - Contains the weights and biases in a binary file

Supported Layers

Layers supported for direct conversion from supported framework layers to intermediate representation layers through the Model Optimizer. While nearly every layer you will ever use is in the supported frameworks is supported, there is sometimes a need for handling Custom Layers.

Custom Layers

Custom layers are those outside of the list of known, supported layers, and are typically a rare exception. Handling custom layers in a neural network for use with the Model Optimizer depends somewhat on the framework used; other than adding the custom layer as an extension, you otherwise have to follow instructions specific to the framework.

Lesson 4

Inference Engine

Provides a library of computer vision functions, supports calls to other computer vision libraries such as OpenCV and performs optimized inference on Intermediate Representation models. Works with various plugins specific to different hardware to support even further optimizations.

Synchronous

Such requests wait for a given request to be fulfilled prior to continuing on to the next request.

Asynchronous

Such requests can happen simultaneously so that the start of the next request does not need to wait for the completion of the previous.

IECore

The main Python wrapper for working with the Inference Engine. Also used to load an `IENetwork`, check the supported layers of a given network, as well as add any necessary CPU extensions.

IENetwork

A class to hold a model loaded from an Intermediate Representation (IR). This can then be loaded into an `IECore` and returned as an `Executable Network`.

ExecutableNetwork

An instance of a network loaded into an `IECore` and ready for inference. It is capable of both synchronous and asynchronous requests and holds a tuple of `InferRequest` objects.

InferRequest

Individual inference requests, such as image by image, to the Inference Engine. Each of these contain their inputs as well as the outputs of the inference request once complete.

Lesson 5

OpenCV

A computer vision (CV) library filled with many different computer vision functions and other useful image and video processing and handling capabilities.

MQTT

A publisher-subscriber protocol often used for IoT devices due to its lightweight nature. The paho-mqtt library is a common way of working with MQTT in Python.

Publish-Subscribe Architecture

A messaging architecture whereby it is made up of publishers, that send messages to some central broker, without knowing of the subscribers themselves. These messages can be posted on some given "topic", which the subscribers can then listen to without having to know the publisher itself, just the "topic".

Publisher

In a publish-subscribe architecture, the entity that is sending data to a broker on a certain "topic".

Subscriber

In a publish-subscribe architecture, the entity that is listening to data on a certain "topic" from a broker.

Topic

In a publish-subscribe architecture, data is published to a given topic, and subscribers to that topic can then receive that data.

FFmpeg

Software that can help convert or stream audio and video. In the course, the related `ffserver` software is used to stream to a web server, which can then be queried by a Node server for viewing in a web browser.

Flask

A Python framework useful for web development and another potential option for video streaming to a web browser.

Node Server

A web server built with Node.js that can handle HTTP requests and/or serve up a webpage for viewing in a browser.