# Intel® Edge AI for IoT Developers Nanodegree Syllabus

## Contact Info

While going through the program, if you have questions about anything, you can reach us at support@udacity.com. For help from Udacity Mentors and your peers visit the Udacity Classroom.

## Nanodegree Program Info

A graduate of this program will be able to: • Leverage the Intel® OpenVINO™ toolkit to fast-track development of high-performance computer vision and deep learning inference applications. • Run pre-trained deep learning models for computer vision on-prem. • Identify key hardware specifications of various hardware types (CPU, VPU, FPGA, and Integrated GPU). • Utilize DevCloud to test model performance on various hardware types (CPU, VPU, FPGA, and Integrated GPU). This program consists of 3 courses and 3 projects. Each project you build will be an opportunity to demonstrate what you've learned in the course, and will demonstrate to potential employers that you have skills in these areas.

### Prerequisite Skills

A well-prepared learner is able to:

- Intermediate knowledge of programming in Python
- Experience with training and deploying deep learning models
- Familiarity with different DL layers and architectures (CNN based)
- Familiarity with the command line (bash terminal)
- Experience using OpenCV

### Required Software

- OpenVino Toolkit 2019
- OpenVino Toolkit 2020.1
- DL Workbench 2020.1
- Intel DevCloud (Optional)

**Version**: 1.0.0

**Length of Program**: 72 Days*

** This is a self-paced program and the length is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. Actual hours may vary.*

# Part 1: Welcome to the Program

Before diving into the program, let's first find out more about what taking a Udacity Nanodegree program is like and what it takes to succeed. We'll also cover software requirements and pre-requisites.

# Part 2: Edge AI Fundamentals with OpenVINO™

Welcome to Edge AI Fundamentals with OpenVINO™, where you'll learn about the basics of AI at the Edge, leverage pre-trained models available with the Intel® Distribution of OpenVINO Toolkit™, convert and optimize other models with the Model Optimizer, and perform inference with the Inference Engine. Additionally, you'll learn some additional topics for edge applications, like MQTT and how to stream video to servers.

### Project: Deploy a People Counter App at the Edge

Show off your new skills by deploying a people counter app at the edge! Go through the full OpenVINO™ Toolkit workflow, along with handling video streams and sending data with MQTT to build your app.

#### Supporting Lessons

| Lesson | Summary |
|---|---|
| **Introduction to AI at the Edge** | Get introduced to AI at the Edge, and find out about the topics you'll learn throughout the rest of the course. |
| **Leveraging Pre-Trained Models** | Utilize Pre-Trained Models from the Intel® Distribution of OpenVINO™ Toolkit to build powerful edge applications, without the need to train your own model. |
| **The Model Optimizer** | Explore the Model Optimizer, which allows you to take models trained with many different Deep Learning frameworks and create an Intermediate Representation useful with the Inference Engine. |
| **The Inference Engine** | Dive deeper into the Inference Engine, and perform inference in the OpenVINO™ Toolkit. By the end, you'll know the full workflow for OpenVINO™ fundamentals and be ready to integrate into an app. |
| **Deploying an Edge App** | With the OpenVINO™ Toolkit fundamentals down, you're ready to move onto more topics to get your edge app up and running. Learn about handling input streams, MQTT and more as you finish the course! |

### Project: Optimize Your GitHub Profile

Other professionals are collaborating on GitHub and growing their network. Submit your profile to ensure your profile is on par with leaders in your field.

# Part 3: Choosing the Right Hardware

Grow your expertise in choosing the right hardware. Identify key hardware specifications of various hardware types (CPU, VPU, FPGA, and Integrated GPU). Utilize the DevCloud to test model performance and deploy power-efficient deep neural network inference on on the various hardware types. Finally, you will distribute workload on available compute devices in order to improve model performance.

### Project: Smart Queuing System

Given real-world scenarios to build a queuing system, use your knowledge of hardware specifications to identify which hardware types work best, and then test the application using the Intel® DevCloud.

#### Supporting Lessons

| Lesson | Summary |
|---|---|
| **Introduction to Hardware at the Edge** | Learn about the key hardware types used on IoT devices and the basics of designing an Edge AI system. |
| **CPUs and Integrated GPUs** | Use the Intel® Devcloud for the Edge for running deep learning models on the CPU and Integrated GPU. |
| **VPUs** | Identify the key specifications of Intel® VPUs and use the Intel® DevCloud for the Edge to run deep learning models on the VPU. |
| **FPGAs** | Identify the key specifications of Intel® FPGAs and utilize the HETERO Plugin to enable efficient hardware utilization. |

### Project: Improve Your LinkedIn Profile

Find your next job or connect with industry peers on LinkedIn. Ensure your profile attracts relevant leads that will grow your professional network.

# Part 4: Optimization Techniques and Tools

Learn how to optimize your model and application code to reduce inference time when running your model at the edge. Use different software optimization techniques to improve the inference time of your model. Calculate how computationally expensive your model is. Use DL Workbench to optimize your model and benchmark the performance of your model. Use a VTune amplifier to find and fix hotspots in your application

code. Finally, package your application code and data so that it can be easily deployed to multiple devices.

## Project: Computer Pointer Controller

Demonstrate your ability to run multiple models in the OpenVINO toolkit on the same machine to control your computer pointer using your eye gaze.

### Supporting Lessons

| Lesson | Summary |
| --- | --- |
| **Introduction to Software Optimization** | This lesson introduces software optimization (SO), when and why you should use it, and the broad classes of SO algorithms |
| **Reducing Model Operations** | Implement optimization techniques that improve performance by reducing the number of model operations. |
| **Reducing Model Size** | One of the ways to perform software optimization is to reduce the size of the model. In this lesson you will learn about the different algorithms used to do that. |
| **Other Optimization Tools and Techniques** | Use VTune Amplifier to measure hotspots in your application code. |

Udacity

Generated Fri Jul 10 12:08:07 PDT 2020