UDACITY

NANODEGREE PROGRAM SYLLABUS

# Intel® Edge AI for IoT Developers

# Overview

Edge AI applications are revolutionizing the IoT industry by bringing fast, intelligent behavior to the locations where it is needed. In this Nanodegree program, you will learn how to develop and optimize Edge AI systems, using Intel's® OpenVINO™ toolkit. 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).

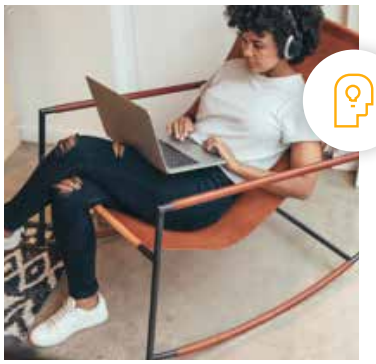
This program consists of 3 courses and 3 projects. Each project you build will be an opportunity to demonstrate what you've learned in the course, and will demonstrate to potential employers that you have skills in these areas.This program is comprised of 4 courses and 4 projects. Each project you build will be an opportunity to demonstrate what you've learned in the lesson, and will demonstrate to potential employers that you have skills in these areas.

IN COLLABORATION WITH

intel

**Estimated Time**: 3 Months at 10 hours / week

**Prerequisites**: Basic Python or C++ experience. Basic familiarity with computer vision and AI model creation.



**Flexible Learning**: Self-paced, so you can learn on the schedule that works best for you.

**Need Help?** udacity.com/advisor Discuss this program with an enrollment advisor.

# Edge AI Fundamentals with OpenVINO™

Leverage a pre-trained model for computer vision inferencing. You will convert pre-trained models into the framework agnostic intermediate representation with the Model Optimizer, and perform efficient inference on deep learning models through the hardware-agnostic Inference Engine. Finally, you will deploy an app on the edge, including sending information through MQTT, and analyze model performance and use cases

**Course Project**
Deploy a People Counter App at the Edge

In this project, you will utilize the Intel® Distribution of the OpenVINO™ Toolkit to build a People Counter app for inference at the edge. You will investigate different pre-trained models for person detection, and then convert the best model for optimized inference. The model will be deployed on the edge, such that only data on 1) the number of people in the frame, 2) time those people spent in frame, and 3) the total number of people counted are sent to a web server; inference will be done on the local machine.

You will need to develop a method to compare the performance of their models before and after use of the OpenVINO toolkit for optimization for edge deployment. You will also examine potential use cases for their deployed people counter app.

The goal of this course is to demonstrate knowledge of the tools taught throughout:

- The Intel® Distribution of OpenVINO™ Toolkit
- The Model Optimizer
- The Inference Engine
- Edge applications
- MQTT

| | LEARNING OUTCOMES | |
|---|---|---|
| **LESSON ONE** | **Leveraging Pre-Trained Models** | • Leverage a pre-trained model for computer vision inferencing |
| **LESSON TWO** | **The Model Optimizer** | • Convert pre-trained models into the framework agnostic intermediate representation with the Model Optimizer |

| | | |
|---|---|---|
| **LESSON THREE** | **The Inference Engine** | • Perform efficient inference on deep learning models through the hardware-agnostic Inference Engine |
| **LESSON FOUR** | **Deploying An Edge App** | • Deploy an app on the edge, including sending information through MQTT, and analyze model performance and use cases. |

# Course 2: Hardware for Computer Vision & Deep Learning Application Deployment

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.

**Course Project**
Smart Queuing System

In this project, you will be given a real-world scenario of building a queuing system for three different clients in three different industry sectors. The sectors will consist of retail, manufacturing, and transportation. Each client will have their own set of constraints and requirements. You'll use your knowledge of hardware specifications to identify which hardware types might work, and then you'll test the application using the Intel® DevCloud to see which hardware performs best. Finally, after reviewing your test results and considering the constraints and requirements of the client, you will propose a hardware solution and justify your selection.

The goal of this course is to grow your expertise in choosing the right hardware. Specifically, you will be able to:

- 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).
- Deploy power-efficient deep neural network inference on CPU, VPU, FPGA and integrated GPU.
- Distribute workload on available compute devices in order to improve model performance.

| | LEARNING OUTCOMES | |
|---|---|---|
| **LESSON ONE** | **Introduction to Hardware at the Edge** | • Describe the importance of selecting the right hardware and the process involved in doing so |

| | | |
|---|---|---|
| **LESSON TWO** | **CPU and Integrated GPU** | • Identify the key specifications of Intel® CPUs and Integrated GPUs<br>• Use the Intel® Devcloud for the Edge for running deep learning models on the CPU and Integrated GPU |
| **LESSON THREE** | **Vision Processing Units** | • Identify the key specifications of Intel® VPUs<br>• Use the Intel® DevCloud for the Edge for running deep learning models on the VPU<br>• Use the MULTI Plugin to get more consistent performance |
| **LESSON FOUR** | **Field Programmable Gate Arrays** | • Identify the key specifications of Intel® FPGAs<br>• Use the Intel® DevCloud for the Edge for running deep learning models on the FPGA<br>• Use the HETERO Plugin to enable efficient hardware utilization |

# Course 3: Optimization Techniques and Tools for Computer Vision Deep Learning Applications

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.

**Course Project**
Computer Pointer Controller

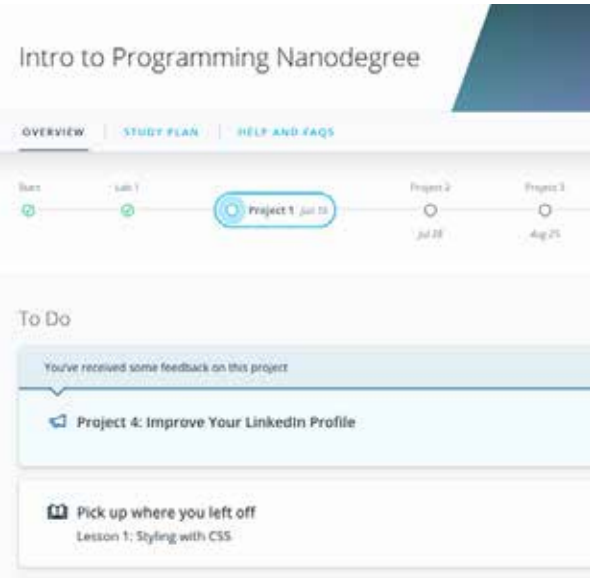
In this project, you will use models available in the OpenVINO toolkit to control your computer pointer using your eye gaze. You will first have to identify faces and extract a face from an input video stream captured from a webcam or a video file. Then you will need to extract facial landmarks and also use a head pose estimation model to find the orientation of the extracted face. Using the head pose and facial landmarks, you will find the orientation of the eye gaze using a gaze estimation model. Finally, you will need to move the mouse pointer in the direction of the eye gaze. This project will demonstrate your ability to run multiple models in the same machine, and coordinate and optimize the flow of data between those models.

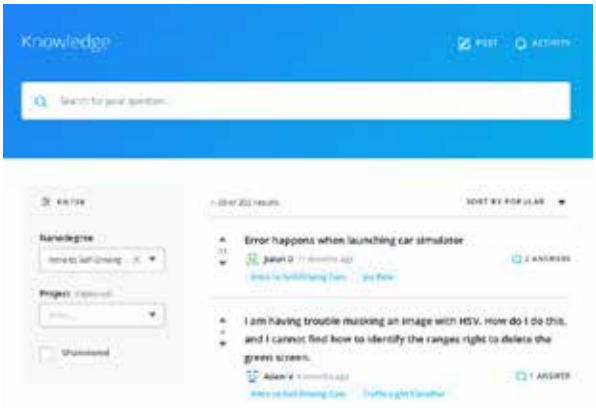
The goal of this course is to help you learn how to optimize your model and application code to reduce inference time when running your model at the edge. Specifically, you will learn how to:

- 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.
- Package your application code and data so that it can be easily deployed to multiple devices.

| | LEARNING OUTCOMES | |
|---|---|---|
| LESSON ONE | **Introduction to Software Optimization** | • Describe why Software Optimization is important<br>• Identify the different fundamental optimization techniques<br>• Use different metrics to measure your model performance<br>• Identify when and how to use optimization techniques |
| LESSON TWO | **Reducing Model Operations** | • Calculate the number of operations in a model<br>• Implement optimization techniques that improve performance by reducing the number of model operations<br>• Use OpenVINO to measure the effects of different optimization techniques on the performance of your model |
| LESSON THREE | **Reducing Model Size** | • Implement optimization techniques that improve performance by reducing the number of model operations<br>• Use DL Workbench to quantize and measure the performance of your model |
| LESSON FOUR | **Other Software Optimization Techniques** | • Use VTune Amplifier to measure hotspots in your application code<br>• Package your application code and data |

# Our Classroom Experience





**REAL-WORLD PROJECTS**
Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

**KNOWLEDGE**
Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover in real-time how to solve the challenges that you encounter.

**STUDENT HUB**
Leverage the power of community through a simple, yet powerful chat interface built within the classroom. Use Student Hub to connect with fellow students in your program as you support and learn from each other.

**WORKSPACES**
See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

**QUIZZES**
Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

**CUSTOM STUDY PLANS**
Preschedule your study times and save them to your personal calendar to create a custom study plan. Program regular reminders to keep track of your progress toward your goals and completion of your program.

**PROGRESS TRACKER**
Stay on track to complete your Nanodegree program with useful milestone reminders.

# Learn with the Best

## Stewart Christie

COMMUNITY MANAGER
AT IOT DEVELOPER PROGRAM

Stewart has been with Intel for the past 20+ years. He is an experienced Technical Marketer with extensive hardware and software knowledge. He has comprehensive understanding of designing embedded electronic hardware and embedded firmware for various applications.

## Michael Virgo

SENIOR CURRICULUM MANAGER
AT UDACITY

After beginning his career in business, Michael utilized Udacity Nanodegree programs to build his technical skills, eventually becoming a Self-Driving Car Engineer at Udacity before switching roles to work on curriculum development for a variety of AI and Autonomous Systems programs.

## Soham Chatterjee

SOFTWARE INNOVATOR
AT INTEL

Soham is an Intel Software Innovator and a former Deep Learning Researcher at Saama Technologies. He is currently a Masters by Research student at NTU, Singapore. His research is on Edge Computing, IoT and Neuromorphic Hardware.

## Archana Iyer

FORMER RESEARCH ENGINEER
AT SAAMA

Archana is a graduate student at the National University of Singapore. She is currently pursuing her research in Deep Learning and Smart Grids, under Professor Dipti Srinivasan. Archana is an Intel Software Innovator and a former Deep Learning Engineer at Saama Technologies.

# All Our Nanodegree Programs Include:

## EXPERIENCED PROJECT REVIEWERS

### REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve

## TECHNICAL MENTOR SUPPORT

### MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions

## PERSONAL CAREER SERVICES

### CAREER SUPPORT

- Resume support
- Github portfolio review
- LinkedIn profile optimization

# Frequently Asked Questions

**PROGRAM OVERVIEW**

### WHY SHOULD I ENROLL?

70% of data being created is at the edge, and only half of that will go to the public cloud; the rest will be stored and processed at the edge, which requires a different kind of developer. Demand for professionals with the Edge AI skills will be immense, as the Edge Artificial Intelligence (AI) software market size is forecasted to grow from $355 Million in 2018, to $1,152 billion by 2023, at an Annual Growth Rate of 27%.**(MarketsandMarkets)** In the Edge AI for IoT Developers Nanodegree program, you'll leverage the potential of edge computing and use the Intel OpenVINO toolkit to fast-track development of high-performance computer vision and deep learning inference applications.

Computer Vision is a fast-growing technology being deployed in nearly every industry from factory floors to amusement parks to shopping malls, smart buildings, and smart homes. It is also driving the evolution of machine learning and human interactions with intelligent systems. Additional applications include drones, security cameras, robots, facial recognition on cell phones, self-driving vehicles, and more, which means these industries and more all need developers with computer vision and deep learning IoT experience.

### WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

This Nanodegree program will prepare you for roles such as IoT Developer, IoT Engineer, Deep Learning Engineer, Machine Learning Engineer, AI Specialist, VPU/CPU/FPGA Developer and more for companies and organizations looking to innovate their hardware on the Edge.

### HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

If you are an enterprise developer and/or professional developer interested in advanced learning, specifically deep learning and computer vision, this program is right for you.

Additionally if you have a background as an IoT Application Prototyper, IoT Application Implementer, IoT System Prototyper, or an IoT System Implementer, or in heterogeneous architectures as an Device Developer, Application Prototyper, Algorithm Developer, Solution Developer, or in security as an Architect/Planner, Security Specialist, or a Protocol Implementer, this program is a good fit.

### WHAT IS EDGE AI? WHAT ARE SOME APPLICATIONS OF THIS TECHNOLOGY?

Edge Computing runs processes locally on the device itself, instead of running them in the cloud. This reduced computing time allows data to be processed much faster, removes the security risk of transferring the data to a cloud-based server, and reduces the cost of data transfer, as well as the risks of bandwidth outages disrupting performance.

# FAQs Continued

Computer vision and AI at the edge are becoming instrumental in powering everything from factory assembly lines and retail inventory management, to hospital urgent care medical imaging equipment like X-ray and CAT scans. Drones, security cameras, robots, facial recognition on cell phones, self-driving vehicles, and more all utilize this technology as well.

According to IEEE Innovation at Work, "By 2020, approximately 20+ billion devices will likely be connected via the Internet of Things (IoT), creating incredible amounts of data every minute. The time it takes to move data to the cloud, perform service on it and then move it back to devices is far too long to meet the increasing needs of the IoT. Unlike cloud computing, which relies on a single data center, edge computing works with a more distributed network, eliminating the round-trip journey to the cloud and offering real-time responsiveness and local authority. It keeps the heaviest traffic and processing closest to the end-user application and devices – smartphones, tablets, home security systems, and more – that generate and consume data. This dramatically reduces latency and leads to real-time, automated decision-making." **(IEEE)**

### WHAT IS THE INTEI® DEVCLOUD FOR THE EDGE?

The Intel® DevCloud for the Edge allows you to actively prototype and experiment with AI workloads for computer vision on Intel hardware.

You have full access to hardware platforms hosted in our cloud environment, designed specifically for deep learning. You can test the performance of your models using the Intel® Distribution of OpenVINO™ Toolkit and combinations of CPUs, GPUs, VPUs such as the Intel® Neural Compute Stick 2 (NCS2) and FPGAs, such as the Intel® Arria® 10. The DevCloud contains a series of Jupyter* notebook tutorials and examples preloaded with everything you needed to quickly get started.

This includes trained models, sample data and executable code from the Intel® Distribution of OpenVINO™ Toolkit as well as other tools for deep learning. These notebooks are designed to help you quickly learn how to implement deep learning applications to enable compelling, high-performance solutions. Intel® has AI hardware waiting for your prototyping of edge inference jobs.

No hardware setup is required on your end. The DevCloud utilizes Jupyter* Notebooks to execute code directly within the Web browser. Jupyter* is a browser-based development environment which allows you to run code and immediately visualize results. You can prototype innovative computer vision solutions in our cloud environment, then execute your code on any of Intel's® available combination of hardware resources.

# FAQs Continued

### WHAT MAKES THE INTEL® EDGE AI FOR IOT DEVELOPERS NANODEGREE PROGRAM UNIQUE?

The Intel® Distribution of OpenVINO™ Toolkit is for developers looking to deploy deep learning models on hardware with Intel chips. Students will be able to interact with Intel's IoT development platform to optimize the performance of their hardware using the DL Workbench. Through Udacity's interactive workspaces, you'll be able to send jobs to Intel'sl® DevCloud and see how different hardware performs in real time.

Deploying AI models on the Edge requires a particular set of tools that providers such as Intel have built. Through Udacity's hands-on exercises that integrate with Intel's platform, students will be able to actually practice testing AI model performance on hardware without needing access to the hardware.

The DevCloud is a cloud-based platform that lets you deploy machine learning models on hardware in the cloud before you purchase the actual hardware so you test and compare the performance of different hardware.

## ENROLLMENT AND ADMISSION

### DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

There is no application. This Nanodegree program accepts everyone, regardless of experience and specific background.

### WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

Prior to entering the program, it is recommended that you have the following knowledge:

- Basic Python or C++ knowledge
- Basics of Computer Vision and AI model creation

### IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

There are a few courses that can help prepare you for the program:

- For Python Experience:**AI Programming with Python** or **Intro to Programming**
- For C++ Experience (this will go in more depth than you will need): **C++**
- For AI Modeling: **Intro to Machine Learning with Pytorch** or **Intro to Machine Learning with TensorFlow**
- For Computer Vision Experience: **Computer Vision**

## TUITION AND TERM OF PROGRAM

### HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The Intel® Edge AI for IoT Developers Nanodegree program is comprised

# FAQs Continued

of content and curriculum to support three (3) projects. We estimate that students can complete the program in three (3) months, working 10 hours per week.

### HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified in the payment card on the Nanodegree program overview page. If you do not graduate within that time period, you will continue learning with month to month payments. See the **Terms of Use** for other policies around the terms of access to our Nanodegree programs.

### I HAVE GRADUATED FROM THE C++ NANODEGREE PROGRAM, WHERE SHOULD I GO FROM HERE?

Many of our graduates continue on to our **Machine Learning Engineer Nanodegree program**, and after that, to the **Self-Driving Car Engineer** and **Artificial Intelligence Nanodegree** programs.

## SOFTWARE AND HARDWARE

### WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

You will need a computer running a 64-bit operating system that has 6th or newer generation of Intel processor running either Win, Ubuntu or (copy from OpenVino).