UDACITY

PROJECT SPECIFICATION

Smart Queue Monitoring System

Proposal

CRITERIA	MEETS SPECIFICATIONS
Extract key information from edge AI scenarios and use this information to select the correct hardware types	For each of the three scenarios, the proposal document indicates the correct hardware and provides valid reasons for choosing that hardware. The reasons must at least include the following: 1. Power Requirements 2. Space Requirements 3. Economic Constraints
Extract key project parameter information from the proposals	Each of the three project proposals includes the following information: 1. Number of people/objects required in the queue 2. Time for processing in the queue 3. Type of Model to be chosen according to device constraints(FP32/FP16/ Int8)

Project Build, Setup and Submission

CRITERIA	MEETS SPECIFICATIONS
Download and implement the correct models from the open model zoo	The code uses the correct model and model precision for each hardware type. The details for the correct model and precision are clearly documented both in the proposal document and deployment IPython notebook.

Provide detailed documentation of the code	The project contains a README file that includes: • Details of the project directory structure • Project setup instructions • Instructions on how to run a simple demo of the project • Detailed documentation of how to run the project for the different scenarios • Documentation of command line arguments (if they are used)
Write code that checks supported model layers before loading a model	For all models used, the code checks supported model layers before loading the model.

Deployment of the Project

CRITERIA	MEETS SPECIFICATIONS
Conduct inference on different edge devices	The project runs models and conducts inference on the following edge devices: • CPU + In. GPU • FPGA • VPU
Use Intel's OpenVino Toolkit to load and run inference on video files	The Jupyter notebook has been completed with the code needed to run inference on the three video files. The notebook: • Includes scripts for downloading the model • Parses arguments to the person_detect.py model file while submitting inference
Run inference on the DevCloud FPGA	The Jupyter notebook contains the following:

	• Average Time taken for inference on FPGA • Time taken to load the model • Command used to submit the job to the DevCloud • Type of Model Precision used • The script file that was submitted to the devcloud The submission also includes the output file generated by devcloud after successful completion of inference job on devcloud
Run inference on the Devcloud CPU and GPU	The Jupyter notebook contains the following: • Average Time taken for inference on CPU and GPU • Time taken to load the model • Command used to submit the job to the DevCloud • Type of Model Precision used • The script file that was submitted to the devcloud The submission also includes the output file generated by devcloud after successful completion of inference job on devcloud
Run inference on the Devcloud VPU	The Jupyter notebook contains the following: • Average Time taken for inference on VPU • Time taken to load the model • Command used to submit the job to the DevCloud • Type of Model Precision used • The script file that was submitted to the devcloud The submission also includes the output file generated by devcloud after successful completion of inference job on devcloud

Suggestions to Make Your Project Stand Out!

1. Use Intel's multiple hardware plugin in your project. Explain how you went about doing it in the README document.

2. Use Intel's Hetero plugin in your project. Explain how you went about doing it in the README document.