Hack2skill Intel OneAPI Hackathon

Slide 1 - Hello and welcome everyone to our demo presentation on oneAPI\_ODAV by team Hitaya.

Slide 2 - Our problem statement is Object Detection for Autonomous Vehicles wherein we propose a solution using state-of-the-art machine learning models.

Our application aims to leverage the power of Intel OneAPI AI Analytics Toolkit and Libraries to develop an advanced object detection and instance segmentation system that enhances the capabilities of autonomous driving to deal with complex problems such as low light vision, adverse road conditions, different weather conditions and stays up to date with latest technology and help users with the tools they need to for real-time object tracking.

Slide 3 – Let us investigate our application architecture design

We’ve implemented a GUI application via python tkinter library. Live camera feed via car dashboard is sent to the object detection & instance segmentation model built on yolov7 and Intel Pytorch for inferencing. This model is trained on COCO dataset which consists of 80 commonly used objects in day-to-day life. We’ve fined tuned this model to our use case and restricted its usage on traffic and road data. On top of this we’ve some more additional data on different weather conditions, adverse road conditions such as potholes, graffiti etc. We’ve implemented these via our data labelling feature which we built from scratch, that allows us to custom annotate coordinates of the bounding boxes. Parallelly we also perform live training over new data received to keep our models updated with unforeseen use-cases.

Slide 4 - Let us now head over to our working POC.

\*\*\*\*\*\*\*\*\*\*\* POC demo \*\*\*\*\*\*\*\*\*

We’re running a GUI app instead of webapp since it can be embedded into car dashboard systems.

Slide 5 – Core components of Intel OneAPI used are the following.

We’re better able to train our models with Intel DevCloud leveraging Intel’s 4th gen Xeon processors. Along with downloadable large open-source datasets on-premise. Model training gains enormous performance speed ups as we make use of Intel extension for PyTorch. On top of these, Intel distribution for Python makes it computationally easier.

Slide 6 – Coming to Results

So evidently, we see greater performance optimizations using Intel’s OneAPI AI analytics toolkit and libraries. Model training gains almost 1.5x performance speed in comparison to normal PyTorch. The graph below shows CPU Wall time comparison between Pytorch vs IPEX.

Slide 7 – Links

Our github codebase links are mentioned below. We shall soon upload our codebase to Intel DevMesh as we were waiting for the hackathon to get over.

Slide 8 - Thank for your time and support throughout the hackathon.

We’re open to questions now.