**Documentation for TensorRT**

TensorRT is developed by NVIDIA for faster inference on NVIDIA GPU and is built on CUDA, NVIDIAs parallel programming model. Its inference is 4 or 5 times faster on any real-time services and 40 times faster compared to CPU performance. There are few major types of optimizations to increase throughput of deep learning models.

The weights or parameters during the model training process are in FP32 precision. By converting into FP16 or INT8 precision through TensorRT, it can significantly reduce the model size and real-time latency although it would affect the accuracy insignificantly. Besides, TensorRT uses layer and tensor fusion to combine all layers with similar input and filter size vertically or horizontally or both to form a single layer in order to optimize the GPU memory and bandwidth.

We cannot directly convert pb files to trt engine, it must convert to onnx first then convert to trt engine.

Step 1: convert pb files to onnx format

* Pip install tf2onnx
* python -m tf2onnx.convert --saved-model ./liho --output liho.onnx
* liho.onnx is generated successfully

Step2: convert onnx format to TRT engine format

* Download and install TensorRT7 on Linux OS and its compatible NVIDIA CUDA version
* /usr/src/tensorrt/bin/trtexec --onnx=Desktop/liho/liho.onnx --shapes=input:1X64X64X3 --saveEngine=li\_engine.trt
* Write a python script to read the image datasets and perform model inference

**Person vs wheelchair classification model**

We perform the model inference for TRT engine using the testing datasets, there are total 320 person images and 320 wheelchair images. Firstly, the image size must be the same so the CV2 is imported and images are resized using [cv2.resize]. I save the numpy arrays into npy format and load it again with the model inference python script. The python script is shown below.



