

## Explaining Custom Layers

The process behind converting custom layers differs, depending on the original model framework. However, in both Tensorflow and Caffe, the first option is to register the custom layers as extensions to the Model Optimizer.

For Tensorflow, the second option is to replace the unsupported subgraph with a different subgraph and offload the computation of the subgraph back to Tensorflow during inference. While for Caffe, is to register the layers as Custom, then use Caffe to calculate the output shape of the layer.

Some of the potential reasons for handling custom layers are as follows:

- If a topology contains any layers that are not in the list of known layers.
- If a device doesn't support a particular layer.

## Comparing Model Performance

Precision benchmark - Using the Openvino benchmark app (Python) :

Hardware Configuration: Intel® Core™ i5-6200U CPU @ 2.30GHz × 4

Start inference asynchronously, 4 inference requests using 4 streams for CPU, inputs: 4 images, limits: 60000 ms duration

	FP32		FP16	
	FPS	Latency	FPS	Latency
ssd_mobilenet_v2	35.11 fps	112.22 ms	35.47 fps	111.29 ms
ssdlite_mobilenet_v2	73.73 fps	50.73 ms	68.89 fps	52.17 ms
person-detection-retail-0013	43.92 fps	86.29 ms	45.19 fps	86.95 ms

Count and Duration:

	FP32		FP16	
	Count	Duration	Count	Duration
ssd_mobilenet_v2	1968 iterations	60163.67 ms	2104 iterations	60156.13 ms
ssdlite_mobilenet_v2	4644 iterations	60054.44 ms	4140 iterations	60099.39 ms
person-detection-retail-0013	2740 iterations	60131.59 ms	2708 iterations	60123.37 ms

Performance Count:

	FP32	FP16
	Total CPU time	Total CPU time
ssd_mobilenet_v2	120535 ms	120991 ms
ssdlite_mobilenet_v2	51253 ms	53476 ms
person-detection-retail-0013	89062 ms	94495 ms

Model speed before converting to Intermediate Representations

Source:

[https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/detection\\_model\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md)

Model	Speed (ms)	COCO mAP[^1]
ssd_mobilenet_v2	31	22
ssdlite_mobilenet_v2	27	22

## Model Size - Pre and Post Conversion:

	Pre conversion	Post conversion	
	Model size	FP32	FP16
ssd_mobilenet_v2	69.7MB	67.3MB	33.6MB
ssdlite_mobilenet_v2	19MB	17.9MB	9.0MB
person-detection-retail-0013		1.4MB	2.9MB

## Assess Model Use Cases

---

Some of the potential use cases of the people counter app are:

- **Museums and Galleries:** Museums, galleries and other public institutions need to be able to accurately measure their footfall as government grants and charitable donations, on which they often rely, are. People counting solutions offer an objective source of reliable data that you can use to study the attraction of your institution. Benefits include direct cost savings by deploying staff only where and when they are needed, identifying new opportunities for bringing in revenue and creating a better environment for your visitors, encouraging even more visits.
- **Shopping Malls:** Shopping Malls draw lots of visitor traffic; not only because of shopping opportunities but also for entertainment value. With people counting, one will be able to see which areas are most crowded and attractive, thus, shape marketing attractions and planning accordingly.
- **Supermarkets:** A supermarket's goal is to turn all visitors into frequent buyers. With people counting data, supermarkets will know how many visitors they get and how many they converted into buyers. People counting information also helps with staff allocation to maximize customer satisfaction.

## Assess Effects on End User Needs

---

Lighting, model accuracy, and camera focal length/image size have different effects on a deployed edge model. The potential effects of each of these are as follows:

- High accuracy, if more resources are available
- Low-power devices will likely have to sacrifice some accuracy for a lighter, faster app, and need some additional considerations about network usage.