

## # Project Write-Up

You can use this document as a template for providing your project write-up. However, if you have a different format you prefer, feel free to use it as long as you answer all required questions.

### ## Explaining Custom Layers

There are supported and unsupported layers of the models by OpenVINO. The unsupported layers are called custom layers. These custom layers should be identified in OpenVINO to handle these layers. This process should be done while converting the model to an Intermediate Representation (IR).

Optimizing and converting a pre-trained model into an IR without a big loss of accuracy is one of the reasons of handling custom layers. The other reason can be determined as improving the performance of the model.

### ## Comparing Model Performance

After a sufficient research, I decided to use the following TensorFlow models.

MODEL 1: `ssd_inception_v2_coco_2018_01_28`:

Converted this model to intermediate representation. This model cannot detect people correctly and therefore it has a very low accuracy. Here you can see the comparison of this model before and after conversion to Intermediate Representation. Inference time and size are slightly better, but accuracy is worse after conversion to IR.

Pre-conversion:

Accuracy: 69.94%

Inference time: 00:06:17

Size: 97.2 MB

Post-conversion:

Accuracy: 63.41%

Inference time: 00:05:09

Size: 95.5 MB

-----

MODEL 2: ssd\_mobilenet\_v2\_coco\_2018\_03\_29:

Converted this model to intermediate representation. This model cannot detect people correctly and therefore it has a relatively low accuracy. Here you can see the comparison of this model before and after conversion to Intermediate Representation. After the conversion, inference time, size and accuracy are all still similar to the pre-conversion model.

Pre-conversion:

Accuracy: 77.91%

Inference time: 00:03:42

Size: 66.4 MB

Post-conversion:

Accuracy: 76.69%

Inference time: 00:03:08

Size: 64.2 MB

-----

MODEL 3: ssdlite\_mobilenet\_v2\_coco\_2018\_05\_09:

Converted this model to intermediate representation. This model cannot detect people correctly and therefore it has a very low accuracy after conversion. Here you can see the comparison of this model before and after conversion to Intermediate Representation. Inference time and size are slightly better, but accuracy is worse after conversion to IR.

Pre-conversion:

Accuracy: 84.43%

Inference time: 00:02:41

Size: 19.7 MB

Post-conversion:

Accuracy: 66.71%

Inference time: 00:02:19

Size: 17.1 MB

-----

All these models perform worse than person-detection-retail-0013 model. Here is the stats for this model. Accuracy and size are way better than the converted models above.

person-detection-retail-0013:

Accuracy: 97.63%

Inference time: 00:02:44

Size: 2.91 MB

### ## Assess Model Use Cases

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

- During these Covid-19 days, this model can be used to get the customer statistics in a marketplace or in a shopping mall.
- Hospitals can use this mode to calculate the spending time of the visitors.

### ## 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:

Lighting is important for the model accuracy that it can detect people more easily.

Camera focal length helps to detect people accurately.

Image size has an effect on the processing time of the image. If the size is bigger, then the processing time will last longer.

## ## Model Research

[This heading is only required if a suitable model was not found after trying out at least three different models. However, you may also use this heading to detail how you converted a successful model.]

In investigating potential people counter models, I tried each of the following three models:

- Model 1: [ssd\_inception\_v2\_coco\_2018\_01\_28]

-

[[https://docs.openvinotoolkit.org/latest/openvino\\_docs\\_MO\\_DG\\_prepare\\_model\\_convert\\_model\\_Convert\\_Model\\_From\\_TensorFlow.html](https://docs.openvinotoolkit.org/latest/openvino_docs_MO_DG_prepare_model_convert_model_Convert_Model_From_TensorFlow.html)]

- I converted the model to an Intermediate Representation with the following arguments:

- wget

[http://download.tensorflow.org/models/object\\_detection/ssd\\_inception\\_v2\\_coco\\_2018\\_01\\_28.tar.gz](http://download.tensorflow.org/models/object_detection/ssd_inception_v2_coco_2018_01_28.tar.gz)

- tar -xvf ssd\_inception\_v2\_coco\_2018\_01\_28.tar.gz

- cd ssd\_inception\_v2\_coco\_2018\_01\_28

- python /opt/intel/openvino/deployment\_tools/model\_optimizer/mo.py --input\_model frozen\_inference\_graph.pb --tensorflow\_object\_detection\_api\_pipeline\_config pipeline.config --reverse\_input\_channels --tensorflow\_use\_custom\_operations\_config /opt/intel/openvino/deployment\_tools/model\_optimizer/extensions/front/tf/ssd\_v2\_support.json

- The model was insufficient for the app because it did not detect second and third person correctly and precisely, thus accuracy was not good for this model. Also inference time of the model was bigger than expected.

- I tried to improve the model for the app by trying different confidence thresholds, but the accuracy did not change significantly.

- Model 2: [ssd\_mobilenet\_v2\_coco\_2018\_03\_29]

- 

[[https://docs.openvinotoolkit.org/latest/openvino\\_docs\\_MO\\_DG\\_prepare\\_model\\_convert\\_model\\_Convert\\_Model\\_From\\_TensorFlow.html](https://docs.openvinotoolkit.org/latest/openvino_docs_MO_DG_prepare_model_convert_model_Convert_Model_From_TensorFlow.html)]

- I converted the model to an Intermediate Representation with the following arguments:

- wget

[http://download.tensorflow.org/models/object\\_detection/ssd\\_mobilenet\\_v2\\_coco\\_2018\\_03\\_29.tar.gz](http://download.tensorflow.org/models/object_detection/ssd_mobilenet_v2_coco_2018_03_29.tar.gz)

- tar -xvf ssd\_mobilenet\_v2\_coco\_2018\_03\_29.tar.gz

- cd ssd\_mobilenet\_v2\_coco\_2018\_03\_29

- python /opt/intel/openvino/deployment\_tools/model\_optimizer/mo.py --input\_model frozen\_inference\_graph.pb --tensorflow\_object\_detection\_api\_pipeline\_config pipeline.config --reverse\_input\_channels --tensorflow\_use\_custom\_operations\_config /opt/intel/openvino/deployment\_tools/model\_optimizer/extensions/front/tf/ssd\_v2\_support.json

- The model was insufficient for the app because the average duration of the detection was not correct for the second person.

- I tried to improve the model for the app by trying different confidence thresholds, it performed better but not good enough to be used.

- Model 3: [ssdlite\_mobilenet\_v2\_coco\_2018\_05\_09]

- 

[[https://docs.openvinotoolkit.org/latest/openvino\\_docs\\_MO\\_DG\\_prepare\\_model\\_convert\\_model\\_Convert\\_Model\\_From\\_TensorFlow.html](https://docs.openvinotoolkit.org/latest/openvino_docs_MO_DG_prepare_model_convert_model_Convert_Model_From_TensorFlow.html)]

- I converted the model to an Intermediate Representation with the following arguments:

- wget

[http://download.tensorflow.org/models/object\\_detection/ssdlite\\_mobilenet\\_v2\\_coco\\_2018\\_05\\_09.tar.gz](http://download.tensorflow.org/models/object_detection/ssdlite_mobilenet_v2_coco_2018_05_09.tar.gz)

- tar -xvf ssdlite\_mobilenet\_v2\_coco\_2018\_05\_09.tar.gz

- cd ssdlite\_mobilenet\_v2\_coco\_2018\_05\_09

- python /opt/intel/openvino/deployment\_tools/model\_optimizer/mo.py --input\_model frozen\_inference\_graph.pb --tensorflow\_object\_detection\_api\_pipeline\_config pipeline.config --

```
reverse_input_channels --tensorflow_use_custom_operations_config  
/opt/intel/opencvino/deployment_tools/model_optimizer/extensions/front/tf/ssd_v2_support.json
```

- The model was insufficient for the app because it detected first three persons for three times, thus accuracy was not good for this model.

- I tried to improve the model for the app by trying different confidence thresholds, but the accuracy did not change significantly.

## ## Conclusion

Second model is an optimum solution among these models in the matter of model accuracy, size and inference time stats. However, all these models did not perform well enough to use efficiently in a necessary case. Therefore, person-detection-retail-0013 model is the best model to be used. It performs way better on the accuracy and the size, and the inference time is not worse than these models.