Exporting a tflite model from a checkpoint

Starting from a trained model checkpoint, creating a tflite model requires 2 steps:

- exporting a tflite frozen graph from a checkpoint
- exporting a tflite model from a frozen graph

Exporting a tflite frozen graph from a checkpoint

With a candidate checkpoint to export, run the following command from tensorflow/models/research:

```
# from tensorflow/models/research
PIPELINE_CONFIG_PATH={path to pipeline config}
TRAINED_CKPT_PREFIX=/{path to model.ckpt}
EXPORT_DIR={path to folder that will be used for export}
python lstm_object_detection/export_tflite_lstd_graph.py \
    --pipeline_config_path ${PIPELINE_CONFIG_PATH} \
    --trained_checkpoint_prefix ${TRAINED_CKPT_PREFIX} \
    --output_directory ${EXPORT_DIR} \
    --add_preprocessing_op
```

After export, you should see the directory ${EXPORT_DIR} containing the following files:

- `tflite_graph.pb`
- `tflite_graph.pbtxt`

Exporting a tflite model from a frozen graph

We then take the exported tflite-compatable tflite model, and convert it to a TFLite FlatBuffer file by running the following:

```
# from tensorflow/models/research
FROZEN_GRAPH_PATH={path to exported tflite_graph.pb}
EXPORT_PATH={path to filename that will be used for export}
PIPELINE_CONFIG_PATH={path to pipeline config}
python lstm_object_detection/export_tflite_lstd_model.py \
       --export_path ${EXPORT_PATH} \
       --frozen_graph_path ${FROZEN_GRAPH_PATH} \
       --pipeline_config_path ${PIPELINE_CONFIG_PATH}
```

After export, you should see the file ${EXPORT_PATH} containing the FlatBuffer model to be used by an application.