Post Training Quantization

- Post-training quantization is a conversion technique that can reduce model size while also improving CPU and hardware accelerator latency, with little degradation in model accuracy.
- One can quantize an already-trained float TensorFlow model when its converted to TensorFlow Lite format using the TensorFlow Lite Converter.
- We will reduce the size of a floating point model by quantizing the weights to float16, it can virtually reduce the model size by half by converting all the float32 weights to float16.

In [ ]:
```
import os
data_dir = "gender_classification_dataset"
if not os.path.isdir(data_dir):
    from google.colab import drive
    drive.mount("/content/drive")
    !cp "/content/drive/MyDrive/Myntra/gender_classification_dataset.zip" "/content/"
    !unzip "gender_classification_dataset.zip"
    !rm "gender_classification_dataset.zip"
    !mv "data/" "gender_classification_dataset/"
    !mv "men/" "gender_classification_dataset/men/"
    !mv "women/" "gender_classification_dataset/women/"
    !cp "/content/drive/MyDrive/Myntra/gender_classification_model.h5" "/content"
    print("Data Loaded Sucessfully!")
else:
    print("Data already loaded!")
```

In [1]:
```
from tensorflow.keras.models import load_model
import tensorflow as tf
from sklearn.metrics import accuracy_score
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
TARGET_SHAPE = (224, 224, 3)
BATCH_SIZE = 200
test_dataset = tf.keras.preprocessing.image_dataset_from_directory(data_dir,validation_split=0.2,subset="validation",seed=123,image_size=TARG
```

```
Found 6660 files belonging to 2 classes.
Using 1332 files for validation.
```

In [4]:
```
def get_file_size(file_path):
    size = os.path.getsize(file_path)
    return size

def convert_bytes(size, unit=None):
    if unit == "KB":
        return print('File size: ' + str(round(size / 1024, 3)) + ' Kilobytes')
    elif unit == "MB":
        return print('File size: ' + str(round(size / (1024 * 1024), 3)) + ' Megabytes')
    else:
        return print('File size: ' + str(size) + ' bytes')
```

In [5]:
```
gender_classifier = load_model("gender_classification_model.h5")
convert_bytes(get_file_size("gender_classification_model.h5"), "MB")
```

```
File size: 679.281 Megabytes
```

In [6]:
```
for X, y in test_dataset.as_numpy_iterator():
    break
```

In [7]:
```
%%time
prediction = gender_classifier.predict(X)
```

```
CPU times: user 59.6 s, sys: 534 ms, total: 1min
Wall time: 31.7 s
```

In [8]:
```
prediction = [1 if x > 0.5 else 0 for x in prediction.reshape(-1,)]
accuracy = accuracy_score(prediction, y)
```

In [9]:
```
print("Inference time for 200 inputs (CPU) : 31.7 seconds")
print("Keras Model Accuarcy : ", round(accuracy * 100, 2), "%")
```

```
Inference time for 200 inputs (CPU) : 31.7 seconds
Keras Model Accuarcy :  98.5 %
```

In [10]:
```
tf_lite_converter = tf.lite.TFLiteConverter.from_keras_model(gender_classifier)
tf_lite_converter.optimizations = [tf.lite.Optimize.OPTIMIZE_FOR_SIZE]
tf_lite_converter.target_spec.supported_types = [tf.float16]
gender_classifier_tflite_model = tf_lite_converter.convert()
open("gender_classifier.tflite", "wb").write(gender_classifier_tflite_model)
convert_bytes(get_file_size("gender_classifier.tflite"), "MB")
```

```
INFO:tensorflow:Assets written to: /tmp/tmp_ua26g98/assets
File size: 142.982 Megabytes
```

In [11]:
```
interpreter = tf.lite.Interpreter(model_path = "gender_classifier.tflite")
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
print("\nBefore Resizing\n")
print("Input Shape:", input_details[0]['shape'])
print("Input Type:", input_details[0]['dtype'])
print("Output Shape:", output_details[0]['shape'])
print("Output Type:", output_details[0]['dtype'])

interpreter.resize_tensor_input(input_details[0]['index'], (200, 224, 224, 3))
interpreter.resize_tensor_input(output_details[0]['index'], (200, 1))
interpreter.allocate_tensors()
```

```
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
print("\nAfter Resizing\n")
print("Input Shape:", input_details[0]['shape'])
print("Input Type:", input_details[0]['dtype'])
print("Output Shape:", output_details[0]['shape'])
print("Output Type:", output_details[0]['dtype'])
```

```
Before Resizing

Input Shape: [  1 224 224   3]
Input Type: <class 'numpy.float32'>
Output Shape: [1 1]
Output Type: <class 'numpy.float32'>

After Resizing

Input Shape: [200 224 224   3]
Input Type: <class 'numpy.float32'>
Output Shape: [200   1]
Output Type: <class 'numpy.float32'>
```

In [12]:
```
%%time
interpreter.set_tensor(input_details[0]['index'], X)
interpreter.invoke()
tflite_model_predictions = interpreter.get_tensor(output_details[0]['index'])
```

```
CPU times: user 54.2 s, sys: 1.02 s, total: 55.2 s
Wall time: 30.2 s
```

In [13]:
```
tflite_model_predictions = [1 if x > 0.5 else 0 for x in tflite_model_predictions.reshape(-1,)]
acc = accuracy_score(tflite_model_predictions, y)
```

In [14]:
```
print("Inference time for 200 inputs (CPU) : 30.2 seconds")
print("TFLite Model Accuarcy : ", round(acc * 100, 2), "%")
```

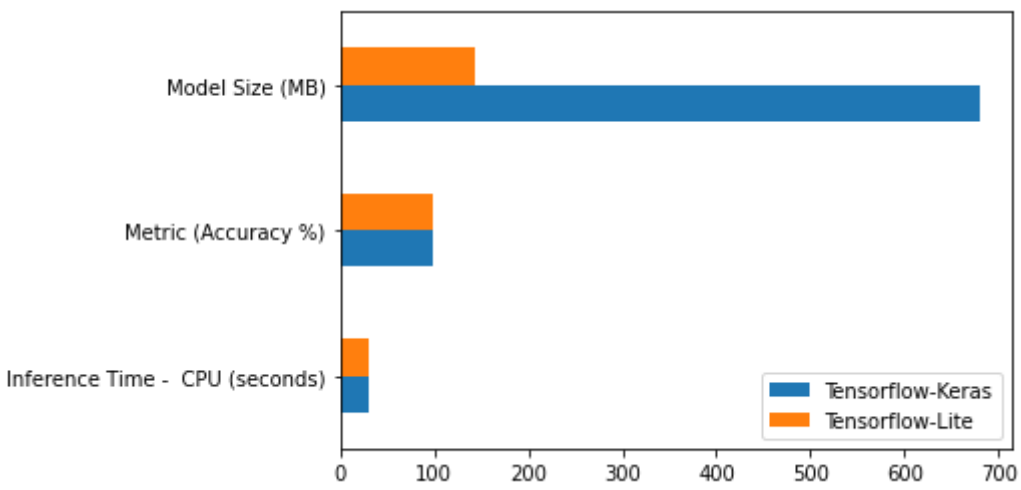
```
Inference time for 200 inputs (CPU) : 30.2 seconds
TFLite Model Accuarcy :  98.5 %
```

In [20]:
```
from tabulate import tabulate
summary = dict()
summary["Model"] = ["Tensorflow-Keras", "Tensorflow-Lite"]
summary["Inference Time -  CPU (seconds)"] = [30.7, 30.2]
summary["Metric (Accuracy %)"] = [98.5, 98.5]
summary["Model Size (MB)"] = [679.281, 142.982]
print("Post Training Quantization Summary for Gender Classification Model: ")
print(tabulate(summary, headers="keys", tablefmt='fancy_grid'))
```

```
Post Training Quantization Summary for Gender Classification Model:
```

Model	Inference Time - CPU (seconds)	Metric (Accuracy %)	Model Size (MB)
Tensorflow-Keras	30.7	98.5	679.281
Tensorflow-Lite	30.2	98.5	142.982

In [21]:
```
summary = pd.DataFrame(summary, index=["Tensorflow-Keras", "Tensorflow-Lite"]).drop(["Model"], axis=1)
summary.T.plot(kind="barh")
plt.show()
```



- We observed no degradation in accuracy metrics and the inference time for the model after quantization.
- However we were able to reduce the model size to 20% of the original size which was expected when choosing the Float16 Quantization.

In [17]:
```
!cp "/content/gender_classifier.tflite" "/content/drive/MyDrive/Myntra"
```