✓ **Congratulations! You passed!**

Grade received 100%   **To pass** 80% or higher

[ **Go to next item** ]

# Week 2 Quiz

Latest Submission Grade 100%

1. To what file do you add the tensorflow lite dependency when building an Android app?   `1 / 1 point`

   ◉ build.gradle

   ○ build.aar

   ○ aar.gradle

   ○ gradle.build

   ✓ **Correct**

2. If the Android Neural networks API is available and you want to use it, how would you do that?   `1 / 1 point`

   ○ You can't use the neural networks API with a TensorFlow Lite model

   ○ Invoke the NNAPI object, and pass the tflite interpreter to it

   ◉ Call the setUseNNAPI method on the interpreter and set its parameter to true

   ○ Do nothing, it will work automatically

   ✓ **Correct**

3. If you want to configure the number of threads the interpreter uses, how would you do that?   `1 / 1 point`

   ○ Do nothing, it's always single threaded

   ○ Do nothing, it automatically picks the appropriate number of threads

   ○ Call the useThreads() method, and it will apportion the correct number of threads

   ◉ Call setNumThreads and pass it the number of threads you want to use

   ✓ **Correct**

4. Where's the best place in an Android app to keep your model?   `1 / 1 point`

   ○ You don't keep your model in your android App, it should download it at runtime

   ○ In the same folder as the activity that calls it

   ○ In the resources folder

   ◉ It can really be anywhere, but for consistency use the assets folder

   ✓ **Correct**

5. If you tested your converted model and know its valid, but the interpreter cannot load it at runtime on Android, what's the most likely reason?   `1 / 1 point`

   ◉ You didn't specify that the model should not be compressed in the build.gradle file

   ○ You have't converted the model to Java or Kotlin format

   ○ You haven't quantized your model

○ You converted your model to iOS format by accident

✓ **Correct**

---

6. What is the method signature of the interpreter when you want to do inference?　　　**1 / 1 point**

○ predicitons = interpreter.predict(inputs)

○ predictions = interpreter.run(inputs)

○ interpreter.predict(inputs, predictions)

◉ interpreter.run(inputs, predictions)

✓ **Correct**

---

7. What Android data structure is most commonly used to feed image input to the interpreter?　　　**1 / 1 point**

○ A TensorArray

○ An Array

◉ A ByteBuffer

○ A Tensor

✓ **Correct**

---

8. How many classes of object can a model trained on the COCO dataset recognize?　　　**1 / 1 point**

◉ 80

○ 1000

○ 10

○ 800

✓ **Correct**

---

9. When performing object recognition, how many dimensions of output tensors are there?　　　**1 / 1 point**

○ 80

◉ 4

○ 10

○ 1

✓ **Correct**

---

10. How do you get the coordinates of the bounding boxes from the object detection model?　　　**1 / 1 point**

○ The coordinates are in the first tensor, read them and simply plot

○ The coordinates are in the first four tensors, read them and simply plot

○ The coordinates are in tensors 0, 1, 2 and 3

◉ The coordinates are in the first tensor, but arranged differently, you have to sort them before you can plot them

✓ **Correct**