

Mobile Engineering: Machine Learning - Musterlösung

Aufgabe 1:

a-c) Fertige runObjectDetection-Funktion:

```
private fun runObjectDetection(bitmap: Bitmap) {
    val image = InputImage.fromBitmap(bitmap, 0)

    // Multiple object detection in static images
    val customObjectDetectorOptions =
        CustomObjectDetectorOptions.Builder(localModel)
            .setDetectorMode(CustomObjectDetectorOptions.SINGLE_IMAGE_MODE)
            .enableMultipleObjects()
            .enableClassification()
            .build()

    val objectDetector =
        ObjectDetection.getClient(customObjectDetectorOptions)

    objectDetector.process(image).addOnSuccessListener { results ->
        debugPrint(results)

        // Parse ML Kit's DetectedObject and create corresponding visualization data
        val detectedObjects = results.map {
            var text = "Unknown"

            // We will show the top confident detection result if it exist
            if (it.labels.isNotEmpty()) {
                val firstLabel = it.labels.first()
                text = "${firstLabel.text}, ${firstLabel.confidence.times(100).toInt()}%"
            }
            BoxWithText(it.boundingBox, text)
        }

        // Draw the detection result on the input bitmap
        val visualizedResult = drawDetectionResult(bitmap, detectedObjects)

        // Show the detection result on the app screen
        imageView.setImageBitmap(visualizedResult)
    }.addOnFailureListener {
        Log.e(TAG, it.message.toString())
    }
}
```

d) Das eingestellte Model erkennt nur einige wenige Labels, unter anderem: Food, Place, Home Object, Plant, Unknown. Objekte im Hintergrund bzw. Objekte auf Bildern mit vielen anderen Objekten werden oft nicht erkannt.

Aufgabe 2:

a) —

b)

```
dependencies {  
    // ...  
    // Object detection & tracking feature with custom bundled model  
    implementation 'com.google.mlkit:object-detection-custom:17.0.0'  
}
```

```
android {  
    // ...  
    aaptOptions {  
        noCompress "tflite"  
        // or noCompress "lite"  
    }  
}
```

c)

```
val localModel = LocalModel.Builder()  
    .setAssetFilePath("model.tflite")  
    // or .setAbsoluteFilePath(absolute file path to model file)  
    // or .setUri(URI to model file)  
    .build()  
  
val customObjectDetectorOptions =  
    CustomObjectDetectorOptions.Builder(localModel)  
        .setDetectorMode(CustomObjectDetectorOptions.SINGLE_IMAGE_MODE)  
        .enableMultipleObjects()  
        .enableClassification()  
        .setClassificationConfidenceThreshold(0.5f)  
        .setMaxPerObjectLabelCount(3)  
        .build()  
  
val objectDetector =  
    ObjectDetection.getClient(customObjectDetectorOptions)
```

d) In der Musterlösung haben wir Inception v4 verwendet. Dieses Model konnte die vorherigen, sehr generischen Labels weiter aufschlüsseln ("Food" wurde zu "Cucumber"). Auch die Erkennungssicherheit hat sich ein wenig verbessert.