private class ImageAnalyzer(ctx: Context, private val listener: RecognitionListener) : ImageAnalysis.Analyzer { ... // TODO 1: Add class variable TensorFlow Lite Model private val AnimalModel = AnimalModel.newInstance(ctx) ...} override fun analyze(imageProxy: ImageProxy) { ... // TODO 2: Convert Image to Bitmap then to TensorImage val tfImage = TensorImage.fromBitmap(toBitmap(imageProxy)) ...}override fun analyze(imageProxy: ImageProxy) { ... // TODO 3: Process the image using the trained model, sort and pick out the top results val outputs = AnimalModel.process(tfImage) .probabilityAsCategoryList.apply { sortByDescending { it.score } // Sort with highest confidence first }.take(MAX\_RESULT\_DISPLAY) // take the top results ...} override fun analyze(imageProxy: ImageProxy) { ... // TODO 4: Converting the top probability items into a list of recognitions for (output in outputs) { items.add(Recognition(output.label, output.score)) } ...} // START - Placeholder code at the start of the codelab. Comment this block of code out.for (i in 0..MAX\_RESULT\_DISPLAY-1){ items.add(Recognition("Fake label $i", Random.nextFloat()))}// END - Placeholder code at the start of the codelab private class ImageAnalyzer(ctx: Context, private val listener: RecognitionListener) : ImageAnalysis.Analyzer { ... // TODO 1: Add class variable TensorFlow Lite Model // Initializing the AnimalModel by lazy so that it runs in the same thread when the process // method is called. private val AnimalModel: AnimalModel by lazy{ // TODO 6. Optional GPU acceleration val compatList = CompatibilityList() val options = if(compatList.isDelegateSupportedOnThisDevice){ Log.d(TAG, "This device is GPU Compatible ") Model.Options.Builder().setDevice(Model.Device.GPU).build() } else { Log.d(TAG, "This device is GPU Incompatible ") Model.Options.Builder().setNumThreads(4).build() } ...}private class ImageAnalyzer(ctx: Context, private val listener: RecognitionListener) : ImageAnalysis.Analyzer { private val AnimalModel: AnimalModel by lazy{ ... // Initialize the Flower Model AnimalModel.newInstance(ctx, options) }}