# Install QuPath StarDist extension

1. Install latest StarDist extension as instructed by <https://github.com/qupath/qupath-extension-stardist>
   1. Straightfoward, drag and drop into open QuPath window
   2. Regardless of using tensorflow-GPU, tensorflow-CPU, or OpenCV, all StarDist will use this
2. Any scripts developed for 0.2.3 using StarDist should have the import line changed to “import qupath.ext.stardist.StarDist2D”
3. Download .pb models from <https://github.com/qupath/models/tree/main/stardist> , and replace path present in 0.2.3 script with their path

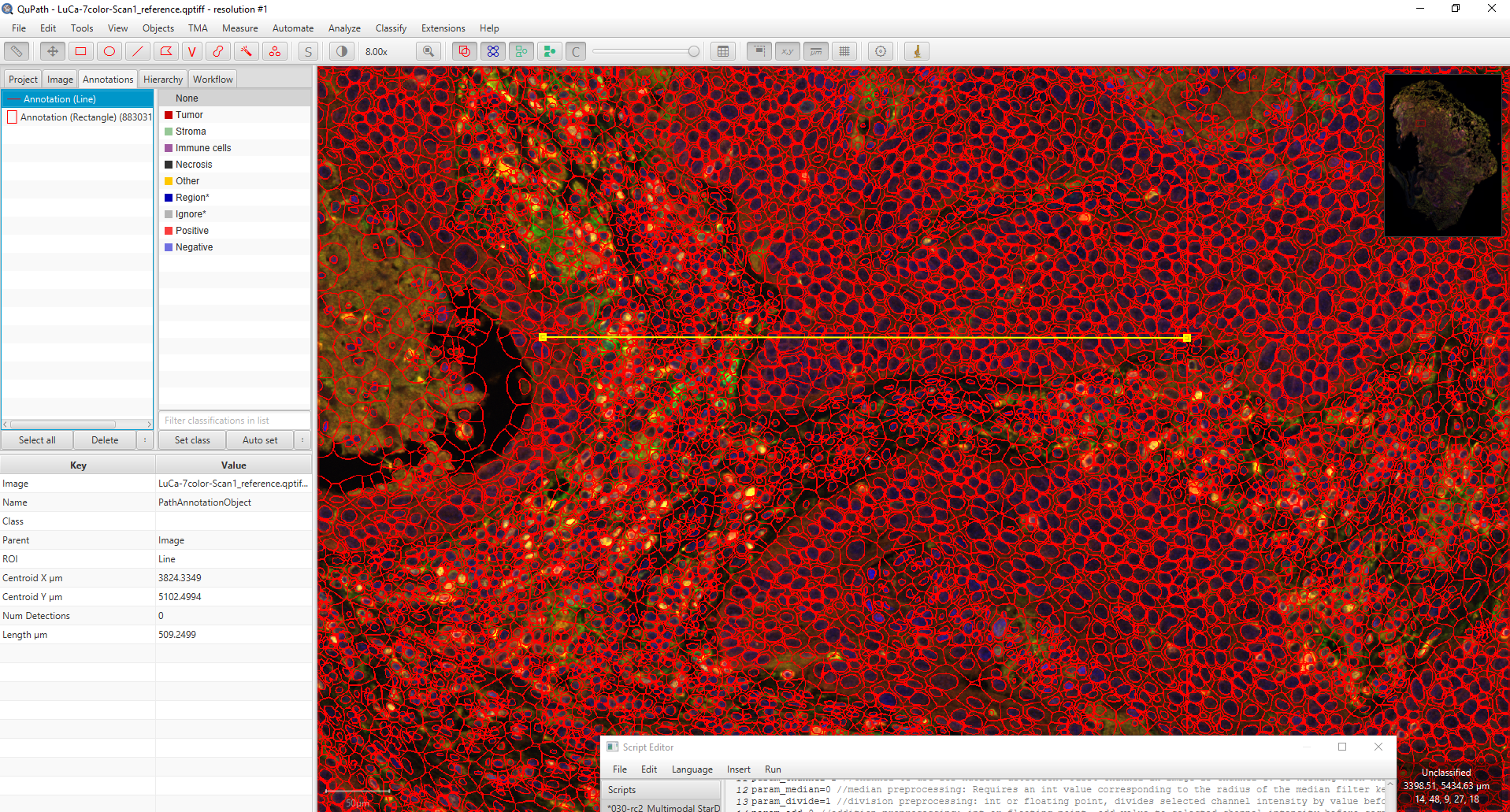
# Tensorflow CPU

1. Download code as .zip from <https://github.com/qupath/qupath-extension-tensorflow> and extract
2. In extracted folder contents, run .\gradlew clean shadowjar
3. In \build\libs drag and drop qupath-extension-tensorflow-0.3.0-rc2-all.jar into QuPath window

# Tensorflow GPU

1. ~~using same code as Tensorflow CPU, run “.\gradlew clean build copyDependencies -P platform=gpu”~~
2. still broken ☹

# Benchmark time for WSI LUCA:

1. 0.3.0-rc2 StarDist OpenCV: 9 minutes, 44.614 seconds (883031 cells)
2. 
3. 0.2.3 StarDist CPU: 9 minutes, 9.022 seconds (880250 cells)
4. 0.3.0-rc2 StarDist Tensorflow-CPU: 10 minutes, 25.502 (883031 cells)

|  |
| --- |
| ERROR: RuntimeException: OpenCV(4.5.3) D:\a\javacpp-presets\javacpp-presets\opencv\cppbuild\windows-x86\_64\opencv-4.5.3\modules\dnn\src\layers\concat\_layer.cpp:102: error: (-201:Incorrect size of input array) Inconsistent shape for ConcatLayer in function 'cv::dnn::ConcatLayerImpl::getMemoryShapes'  ERROR: org.bytedeco.opencv.opencv\_dnn.Net.forward(Native Method)  qupath.opencv.dnn.OpenCVDnn$OpenCVNetFunction.predict(OpenCVDnn.java:690)  qupath.opencv.dnn.OpenCVDnn$OpenCVNetFunction.predict(OpenCVDnn.java:704)  qupath.opencv.dnn.DnnModel.convertAndPredict(DnnModel.java:100)  qupath.ext.stardist.StarDist2D.detectObjectsForTile(StarDist2D.java:1020)  qupath.ext.stardist.StarDist2D.lambda$detectObjects$5(StarDist2D.java:796)  java.base/java.util.stream.ReferencePipeline$7$1.accept(Unknown Source)  java.base/java.util.ArrayList$ArrayListSpliterator.forEachRemaining(Unknown Source)  java.base/java.util.stream.AbstractPipeline.copyInto(Unknown Source)  java.base/java.util.stream.AbstractPipeline.wrapAndCopyInto(Unknown Source)  java.base/java.util.stream.ReduceOps$ReduceTask.doLeaf(Unknown Source)  java.base/java.util.stream.ReduceOps$ReduceTask.doLeaf(Unknown Source)  java.base/java.util.stream.AbstractTask.compute(Unknown Source)  java.base/java.util.concurrent.CountedCompleter.exec(Unknown Source)  java.base/java.util.concurrent.ForkJoinTask.doExec(Unknown Source)  java.base/java.util.concurrent.ForkJoinPool$WorkQueue.topLevelExec(Unknown Source)  java.base/java.util.concurrent.ForkJoinPool.scan(Unknown Source)  java.base/java.util.concurrent.ForkJoinPool.runWorker(Unknown Source)  java.base/java.util.concurrent.ForkJoinWorkerThread.run(Unknown Source) |