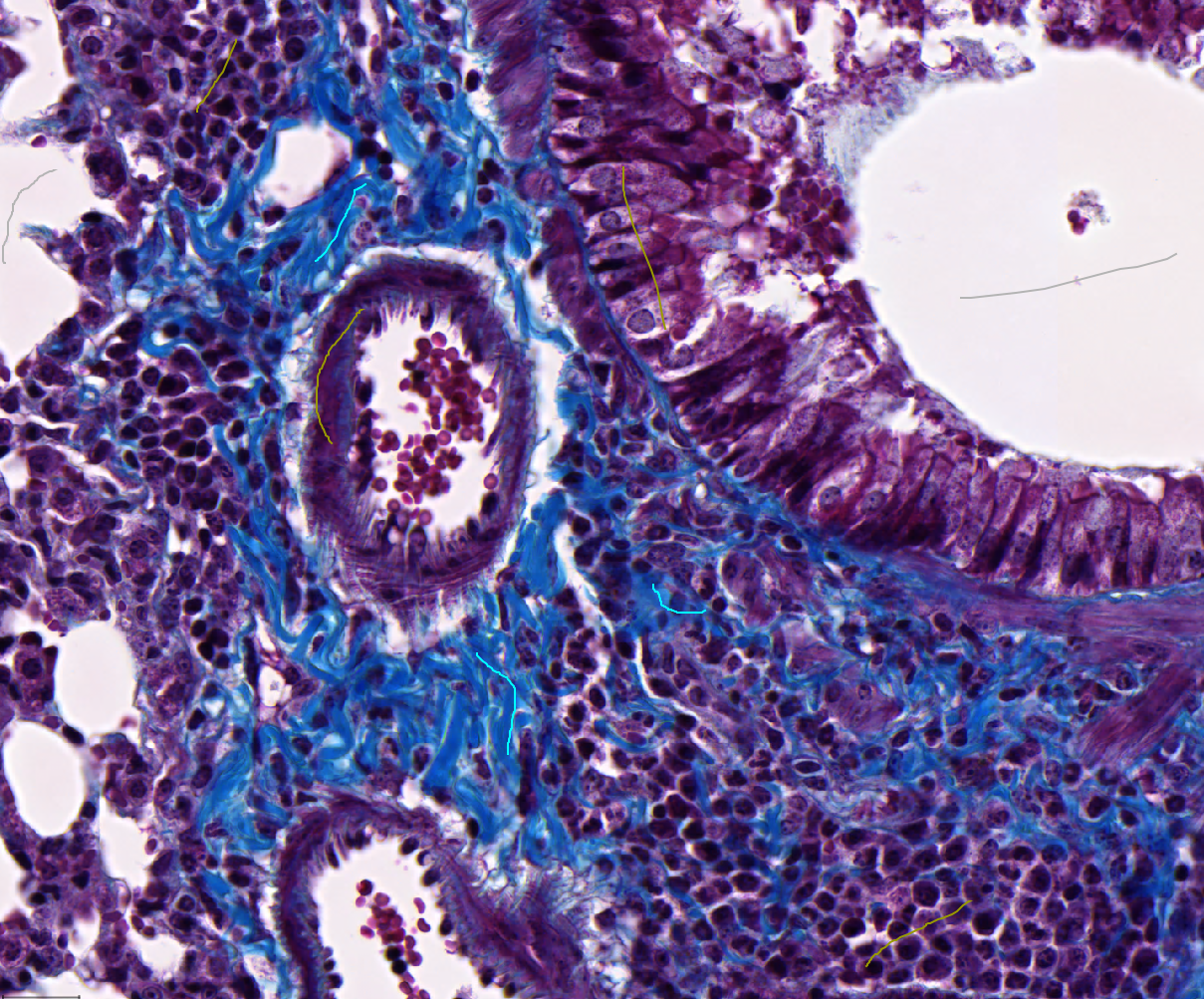
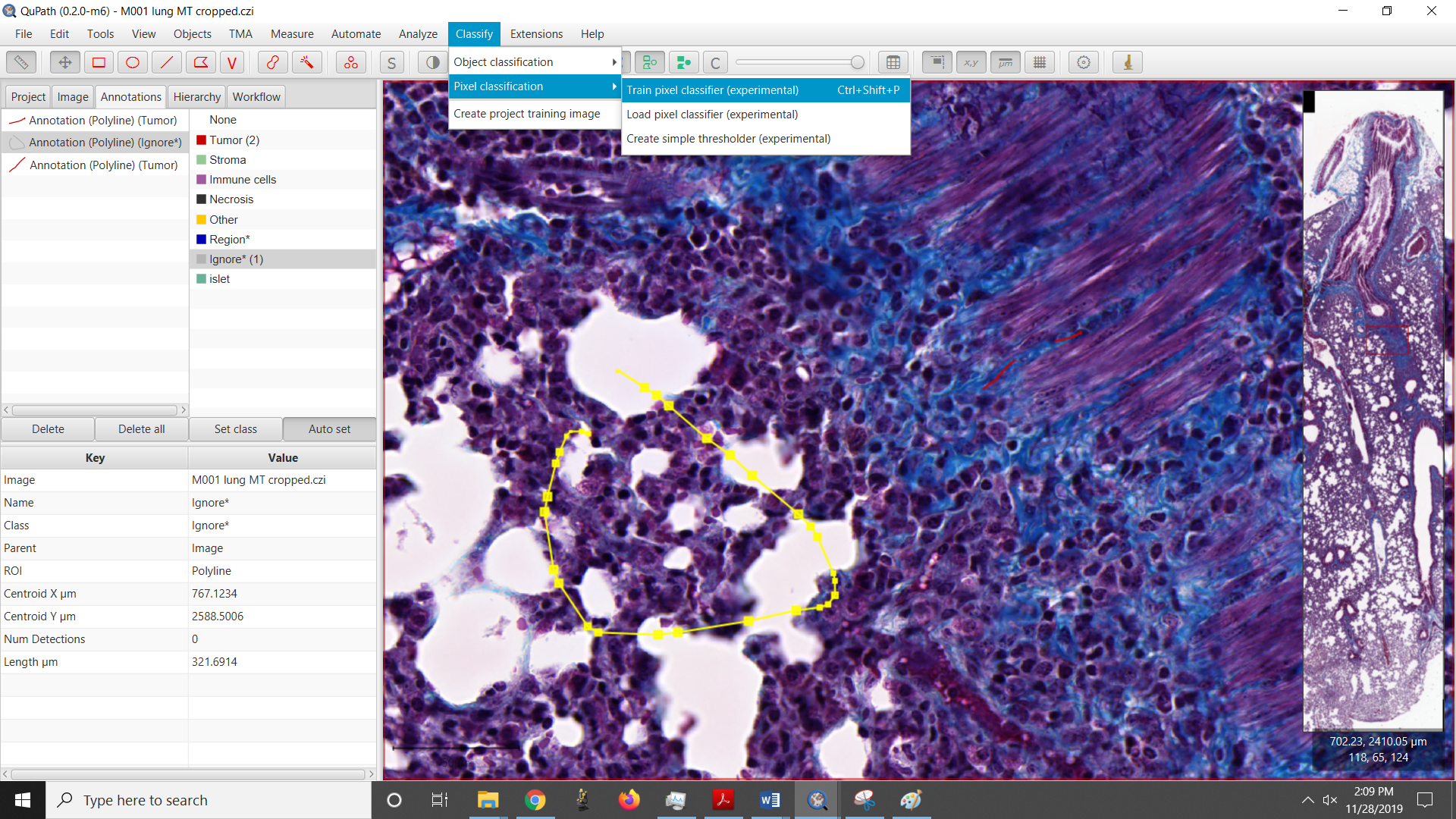
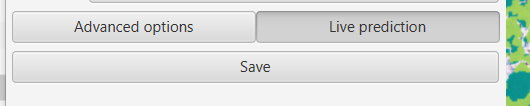
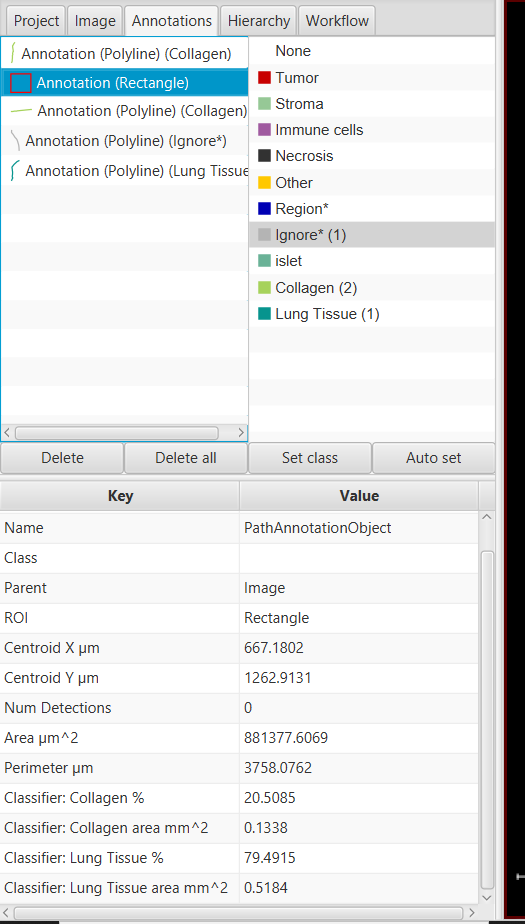
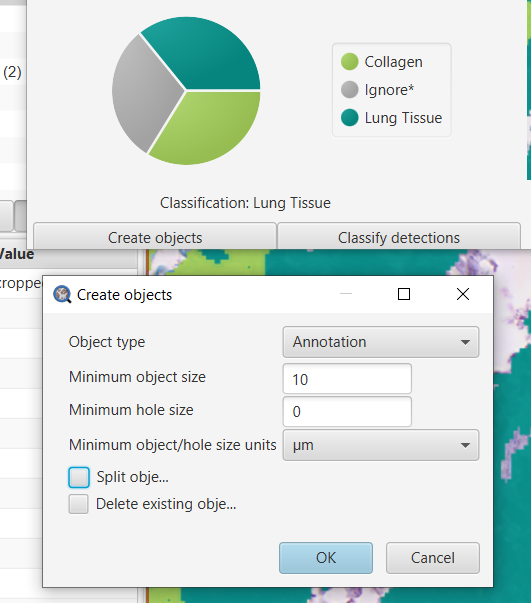
**Measuring Collagen in the Lung**

Step-by-step Protocol

**Goals:** Learn how to use the pixel classifier

Assumes you are already familiar with projects, annotations, etc

1. Use QuPath 0.2.0\_m6
2. Download “M001 Lung MT cropped.czi” and “M007 Lung MT cropped.czi”.  
   These are images of inflamed mouse lungs stained with Masson’s Trichrome, which highlights collagen in blue, nuclei in brown/black and other tissue components in dark red/purple. In healthy lungs, large airways and blood vessels will have a collagen ring surrounding them. During inflammation, the amount of collagen increases and it spreads further from the large airways.
3. Create a new project
   1. Use “Add Images” to add the files to the new project OR drag-and-drop images into the QuPath window
4. Create 2 new classes: “Collagen” and “Lung Tissue”. The specific names are unimportant, just choose ones that are meaningful to you. Don’t use “Region”, “Ignore”, or anything with a star (\*)- those are special. You can change the colors for the classes.
5. Use the polyline tool to begin to annotate regions of collagen, other tissue, or background. Use the “Ignore” class for background. Don’t forget to set each annotation to a particular class. See example below, where bright blue = Collagen; green = Lung Tissue; and grey = Ignore. Start with ~5 lines of each class in different regions of the slide.   
   
6. Create a pixel classifier. Go to Classify > Pixel classification > Train pixel classifier  
     
     
     
   A window will pop up asking you to set some of the parameters of the classifier. Here’s a quick (and incomplete) overview of what the different options mean:
   1. *Classifier*: Random Trees, K-nearest neighbor, Logistic Regression, or Artificial Neural Network. These 4 different algorithms each have their pros and cons, which can be explored at length elsewhere on the internet (for instance, with [Weka](https://www.cs.waikato.ac.nz/ml/weka/)). For this example, we found ANN to work best, but please play with this to get a feel for their different outputs.
   2. *Resolution*: The pixel size of the eventual output of the classification. Lower resolution (larger pixels) calculates faster. The image on the right side of the window demonstrates the chosen pixel size. \*Using full resolution on large images may cause the program to become unstable on weaker computers!\* Draw a small annotation (see point *e* below) and zoom in for a more smooth experience.
   3. *Features:* Which data to use as an input to the classifier, including both raw and processed images. First, select which channels to use (for RGB images, you typically want all channels, but for fluorescence images, you may want to only select relevant channels). The “features” are different filters that are applied to every pixel of the image (gaussian smoothing, median, laplacian of the gaussian, etc) that can be thought of as creating additional channels with ,pre information. You can choose the scales at which these filters are applied. A small scale leads to finer, but noisier, results. All selected features are calculated for all selected scales on all selected channels.
   4. *Output:* The “classification” output overlays the program’s best guess for the correct class on top of each pixel. Every pixel will be assigned 1 class. The Ignore class is not shown. Some classifiers (ANN), can instead output the probability of the most likely class (more probable = more opaque). This is useful for seeing where the classifier is confused so that you can focus your annotations in those regions.
   5. *Region:* Choose whether the live updates of the classifier should be applied to the entire image or just the annotated regions (only applies to annotations with an area, like polygon or rectangle). Use the second option for computers with less resources.
7. Press “live prediction” to see the current classification results. Make sure you have the classifier view turned on. 
8. Inevitably, there will be some errors. Continue adding annotations over the incorrect classifications to fix them. The results will update continuously as long as “live prediction” is pressed. Also, try changing any of the parameters described above to see if those changes will improve the results. Make sure you look at different regions of the image to be sure there are no local issues.
   1. The pie chart in the window shows how many pixels were annotated as each class. Try to keep these approximately balanced (at least, for the Collagen and Lung Tissue classes. The Ignore class can be much smaller or larger).
   2. Users more familiar with machine learning can use the “Advanced Options” to tweak the classifier. Otherwise, stick with the defaults.
9. Once you are satisfied with the results, save the classifier. This will create a subfolder inside the project folder with a file that contains the optimized method for classifying each pixel. It does not save the annotations that you used to train the classifier. With that file, you can reload this exact classifier on other images later, but you cannot add new training objects to it to change how things are classified.   
     
   
10. To process the entire image, unselect all objects by selecting the movement tool  and double clicking anywhere outside of an annotation. If you instead only want to classify a small portion of the image (i.e, an annotation drawn around an interesting piece of the tissue), select that annotation. Lines cannot be selected because they do not have an area.
    1. While live prediction in on, if you select a region, it will display in the property the area inside that region that belong to each class. It will also calculate what fraction of all of the classified pixels (not including Ignore\* or Region\*) belong to each class, e.g. of everything that is not ignored, how much is Collagen and how much is Lung Tissue.
11. Press “Create Objects” (bottom left corner).  
      
      
      
    A window will pop up asking a few questions:
    1. Will the final objects be annotations or detections? For this example, choose annotations.
    2. Very small details can be ignored to make processing significantly faster. You can tell the program what is the minimum area to care about for objects and holes, and whether that is in pixels or microns. In this example, choose to suppress any object smaller than 10 um.
    3. With “Split objects” unchecked, you will get 1 object per class. If it is checked, each region of connected pixels will become an individual object. Leave it unchecked for this application.
    4. You can choose whether or not to delete all objects inside the processed area (not including the region outline for annotations). This choice depends on what came before or after in your workflow. If you are going to delete existing objects, it is a good idea to save them to a text file first, using the scripts written by Svidro [here](https://gist.github.com/Svidro/5829ba53f927e79bb6e370a6a6747cfd#file-copying-annotations-between-images-groovy). For now, do not delete the objects.
12. 2 annotations will be created- one for Lung Tissue and one for Collagen. The “Ignore” class was ignored. Click on each to see and record the total area classified as that type.
13. Repeat for the next image. Or, try your own!

Note:

***Random trees*** classifier uses a random number (seed) to grow the trees during training. QuPath creates a new seed every time you open an image. This will often lead to a different classification result, even if you use the same annotations to train. ***Artificial Neural Networks*** will always give you consistent results.