

# Nerves Segmentation

## Importing libraries

Importing libraries to access directory, to load image, to use array, to plot, to process image, and generate random variables

```
In [1]: import os, cv2, csv, random, time
from pathlib import Path
from os import listdir
from os.path import isfile, join
import openslide
import numpy as np
import matplotlib.pyplot as plt
from skimage.morphology import binary_closing, binary_opening, binary_dilation, b
from skimage.color import rgb2hsb
from skimage.transform import resize
from PIL import Image, ImageDraw
import scipy.misc
from itertools import combinations, permutations

import keras
from keras import backend as K
from keras.utils.data_utils import get_file
from keras.models import Sequential, Model
from keras.layers.core import Flatten, Dense, Dropout, Lambda, Reshape
from keras.layers import Input, LeakyReLU, Dense, Layer, LocallyConnected1D, Activat
from keras.layers.convolutional import Convolution2D, MaxPooling2D, ZeroPadding2D
from keras.layers import Conv1D, Conv2D, Conv3D, Conv2DTranspose, ConvLSTM2D, Simpl
from keras.layers import UpSampling2D, merge, Reshape
from keras.optimizers import SGD, RMSprop, Adam
from keras.preprocessing import image
from keras.layers.normalization import BatchNormalization
from keras import optimizers
from keras import regularizers
from keras.engine import InputLayer
from keras.callbacks import ModelCheckpoint
```

```
C:\Users\Indri92\Anaconda3\envs\tfgpu\lib\site-packages\h5py\_init_.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
    from ._conv import register_converters as _register_converters
Using TensorFlow backend.
```

## Importing Functions

```
In [2]: from utils import getRowsFromCSV, extractBBWithSizeTh, paddedMorphologicalOperation
from utils import filterColourRange, drawPredBoxOnImage
from utils import scoringRoiWrtAnnot, combineBoxes, processCC
from utils import saveImageToJpg, savingCSVFile, boundingBoxOnWSI, saveImageFromRoi

from preprocessing import preprocessListWSI
from model import get_CNN
from predict import printscoresDict, filterListWSI
from directDL import predictDirectDL, directPredictListWSI, directDLResults
```

## Listing Training, Validation and Testing Files

```
In [3]: demoFiles=['10039', '10097', '10114']
print("Number of demo files is "+str(len(demoFiles)))

testingFiles=['10012', '10023', '10029', '10036', '10039', '10049', '10064', '10067',
              '10071', '10072', '10073', '10078', '10087', '10088', '10093',
              '10096', '10097', '10102', '10113', '10114', '10116', '10121']
print("Number of testing files is "+str(len(testingFiles)))
```

Number of demo files is 3  
 Number of testing files is 22

## Defining Path for Source Files

```
In [4]: fileDir=os.path.join(os.getcwd(),"dataFiles")
slicePath=os.path.join(fileDir,"svs files")

#annotFile.csv consists of fileName and bounding box coordinate [x, y, w, h] of the ROI
annotPath=os.path.join(fileDir,"annot csv")

#sArea.csv consists of fileName and bounding box coordinate [x, y, w, h] of the segmented area
sArea=getRowsFromCSV(os.path.join(fileDir,"sArea.csv"),'Dict')
```

## Defining Mean and Std

```
In [5]: pxlToUm=0.5036
xTrainMean=np.array([2.13762123e+02, 2.07595257e+02, 2.03273641e+02])
xTrainStd=np.array([36.67175097, 39.33090933, 42.27675675])
```

## Loading CNN Model (Trained with HandPicked data and shifted augmentation)

```
In [6]: #input pretrained model path
pretrainedModelPath=os.path.join(fileDir,"pretrained models","pretrained_CNN.h5")
#Load CNN model
model=get_CNN((160,160), chN=3, lr = 1e-4, loss='binary_crossentropy', metrics=['accuracy'])
#Displaying model architecture summary
model.summary()
```

D:\UniData\Projects\NervesSegmentationProject\model.py:152: UserWarning: Update your `Model` call to the Keras 2 API: `Model(inputs=Tensor("in...", outputs=[tf.Tensor(..., shape=(None, 160, 160, 3), name='input')])`  
model = Model(input = inputs, output = [finalLayer,fc1c])

Model loaded: D:\UniData\Projects\NervesSegmentationProject\dataFiles\pretrained models\pretrained\_CNN.h5

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 160, 160, 3)	0	
batch_normalization_1 (BatchNor	(None, 160, 160, 3)	12	input_1[0][0]
conv2d_1 (Conv2D)	(None, 160, 160, 16)	448	batch_normalization_1[0][0]
batch_normalization_2 (BatchNor	(None, 160, 160, 16)	64	conv2d_1[0][0]
activation_1 (Activation)	(None, 160, 160, 16)	0	batch_normalization_2[0][0]
conv2d_2 (Conv2D)	(None, 160, 160, 16)	2320	activation_1[0][0]
batch_normalization_3 (BatchNor	(None, 160, 160, 16)	64	conv2d_2[0][0]
activation_2 (Activation)	(None, 160, 160, 16)	0	batch_normalization_3[0][0]
concatenate_1 (Concatenate)	(None, 160, 160, 32)	0	activation_1[0][0]
			activation_2[0][0]
average_pooling2d_1 (AveragePoo	(None, 80, 80, 32)	0	concatenate_1[0][0]
conv2d_3 (Conv2D)	(None, 80, 80, 32)	9248	average_pooling2d_1[0][0]
batch_normalization_4 (BatchNor	(None, 80, 80, 32)	128	conv2d_3[0][0]
activation_3 (Activation)	(None, 80, 80, 32)	0	batch_normalization_4[0][0]
conv2d_4 (Conv2D)	(None, 80, 80, 32)	9248	activation_3[0][0]

[0][0]

batch_normalization_5 (BatchNor (None, 80, 80, 32)	128	conv2d_4[0][0]
activation_4 (Activation) (None, 80, 80, 32)	0	batch_normalization_5[0][0]
concatenate_2 (Concatenate) (None, 80, 80, 64)	0	activation_3 activation_4
average_pooling2d_2 (AveragePoo (None, 40, 40, 64)	0	concatenate_2 [0][0]
conv2d_5 (Conv2D) (None, 40, 40, 64)	36928	average_poolin g2d_2[0][0]
batch_normalization_6 (BatchNor (None, 40, 40, 64)	256	conv2d_5[0][0]
activation_5 (Activation) (None, 40, 40, 64)	0	batch_normaliz ation_6[0][0]
conv2d_6 (Conv2D) (None, 40, 40, 64)	36928	activation_5 [0][0]
batch_normalization_7 (BatchNor (None, 40, 40, 64)	256	conv2d_6[0][0]
activation_6 (Activation) (None, 40, 40, 64)	0	batch_normaliz ation_7[0][0]
concatenate_3 (Concatenate) (None, 40, 40, 128)	0	activation_5 [0][0] activation_6 [0][0]
average_pooling2d_3 (AveragePoo (None, 20, 20, 128)	0	concatenate_3 [0][0]
conv2d_7 (Conv2D) (None, 20, 20, 128)	147584	average_poolin g2d_3[0][0]
batch_normalization_8 (BatchNor (None, 20, 20, 128)	512	conv2d_7[0][0]

activation_7 (Activation) activation_8[0][0]	(None, 20, 20, 128) 0	batch_normalization
conv2d_8 (Conv2D) [0][0]	(None, 20, 20, 128) 147584	activation_7
batch_normalization_9 (BatchNor [0][0])	(None, 20, 20, 128) 512	conv2d_8[0][0]
activation_8 (Activation) activation_9[0][0]	(None, 20, 20, 128) 0	batch_normalization
concatenate_4 (Concatenate) [0][0]	(None, 20, 20, 256) 0	activation_7
		activation_8
[0][0]		
dropout_1 (Dropout) [0][0]	(None, 20, 20, 256) 0	concatenate_4
average_pooling2d_4 (AveragePoo [0])	(None, 10, 10, 256) 0	dropout_1[0]
conv2d_9 (Conv2D) g2d_4[0][0]	(None, 10, 10, 256) 590080	average_poolin
batch_normalization_10 (BatchNo [0][0])	(None, 10, 10, 256) 1024	conv2d_9[0][0]
activation_9 (Activation) activation_10[0][0]	(None, 10, 10, 256) 0	batch_normaliz
conv2d_10 (Conv2D) [0][0]	(None, 10, 10, 256) 590080	activation_9
batch_normalization_11 (BatchNo [0])	(None, 10, 10, 256) 1024	conv2d_10[0]
activation_10 (Activation) activation_11[0][0]	(None, 10, 10, 256) 0	batch_normaliz
concatenate_5 (Concatenate) [0][0]	(None, 10, 10, 512) 0	activation_9
		activation_10

[0][0]

dropout_2 (Dropout) [0][0]	(None, 10, 10, 512) 0	concatenate_5
up_sampling2d_1 (UpSampling2D) [0]	(None, 20, 20, 512) 0	dropout_2[0]
conv2d_12 (Conv2D) 1[0][0]	(None, 20, 20, 128) 262272	up_sampling2d_1
batch_normalization_13 (BatchNo [0]	(None, 20, 20, 128) 512	conv2d_12[0]
activation_12 (Activation) activation_13[0][0]	(None, 20, 20, 128) 0	batch_normalization_13
concatenate_6 (Concatenate) [0][0]	(None, 20, 20, 384) 0	concatenate_4
		activation_12
conv2d_13 (Conv2D) [0][0]	(None, 20, 20, 128) 442496	concatenate_6
batch_normalization_14 (BatchNo [0]	(None, 20, 20, 128) 512	conv2d_13[0]
activation_13 (Activation) activation_14[0][0]	(None, 20, 20, 128) 0	batch_normalization_14
conv2d_14 (Conv2D) [0][0]	(None, 20, 20, 128) 147584	activation_13
batch_normalization_15 (BatchNo [0]	(None, 20, 20, 128) 512	conv2d_14[0]
activation_14 (Activation) activation_15[0][0]	(None, 20, 20, 128) 0	batch_normalization_15
concatenate_7 (Concatenate) [0][0]	(None, 20, 20, 256) 0	activation_13
		activation_14
[0][0]		

up_sampling2d_2 (UpSampling2D)	(None, 40, 40, 256)	0	concatenate_7 [0][0]
conv2d_15 (Conv2D)	(None, 40, 40, 64)	65600	up_sampling2d_2 [0][0]
batch_normalization_16 (BatchNo)	(None, 40, 40, 64)	256	conv2d_15[0] [0]
activation_15 (Activation)	(None, 40, 40, 64)	0	batch_normalization_16[0][0]
concatenate_8 (Concatenate)	(None, 40, 40, 192)	0	concatenate_3 [0][0]
activation_15			[0][0]
conv2d_16 (Conv2D)	(None, 40, 40, 64)	110656	concatenate_8 [0][0]
batch_normalization_17 (BatchNo)	(None, 40, 40, 64)	256	conv2d_16[0] [0]
activation_16 (Activation)	(None, 40, 40, 64)	0	batch_normalization_17[0][0]
conv2d_17 (Conv2D)	(None, 40, 40, 64)	36928	activation_16 [0][0]
batch_normalization_18 (BatchNo)	(None, 40, 40, 64)	256	conv2d_17[0] [0]
activation_17 (Activation)	(None, 40, 40, 64)	0	batch_normalization_18[0][0]
concatenate_9 (Concatenate)	(None, 40, 40, 128)	0	activation_16 [0][0]
activation_17			[0][0]
up_sampling2d_3 (UpSampling2D)	(None, 80, 80, 128)	0	concatenate_9 [0][0]

conv2d_18 (Conv2D) 3[0][0]	(None, 80, 80, 32)	16416	up_sampling2d_
batch_normalization_19 (BatchNo [0]	(None, 80, 80, 32)	128	conv2d_18[0]
activation_18 (Activation) activation_19[0][0]	(None, 80, 80, 32)	0	batch_normaliz
concatenate_10 (Concatenate) [0][0]	(None, 80, 80, 96)	0	concatenate_2
			activation_18
[0][0]			
conv2d_19 (Conv2D) [0][0]	(None, 80, 80, 32)	27680	concatenate_10
batch_normalization_20 (BatchNo [0]	(None, 80, 80, 32)	128	conv2d_19[0]
activation_19 (Activation) activation_20[0][0]	(None, 80, 80, 32)	0	batch_normaliz
conv2d_20 (Conv2D) [0][0]	(None, 80, 80, 32)	9248	activation_19
batch_normalization_21 (BatchNo [0]	(None, 80, 80, 32)	128	conv2d_20[0]
activation_20 (Activation) activation_21[0][0]	(None, 80, 80, 32)	0	batch_normaliz
concatenate_11 (Concatenate) [0][0]	(None, 80, 80, 64)	0	activation_19
			activation_20
[0][0]			
up_sampling2d_4 (UpSampling2D) [0][0]	(None, 160, 160, 64)	0	concatenate_11
conv2d_21 (Conv2D) 4[0][0]	(None, 160, 160, 16)	4112	up_sampling2d_

batch_normalization_22 (BatchNo [0]	(None, 160, 160, 16) 64	conv2d_21[0]
activation_21 (Activation) activation_22[0][0]	(None, 160, 160, 16) 0	batch_normalization_22[0]
concatenate_12 (Concatenate) [0][0]	(None, 160, 160, 48) 0	concatenate_1
activation_21 [0][0]		activation_21
conv2d_22 (Conv2D) [0][0]	(None, 160, 160, 16) 6928	concatenate_12
batch_normalization_23 (BatchNo [0]	(None, 160, 160, 16) 64	conv2d_22[0]
activation_22 (Activation) activation_23[0][0]	(None, 160, 160, 16) 0	batch_normalization_23[0]
conv2d_23 (Conv2D) [0][0]	(None, 160, 160, 16) 2320	activation_22
conv2d_11 (Conv2D) [0]	(None, 10, 10, 8) 4104	dropout_2[0]
batch_normalization_24 (BatchNo [0]	(None, 160, 160, 16) 64	conv2d_23[0]
batch_normalization_12 (BatchNo [0]	(None, 10, 10, 8) 32	conv2d_11[0]
activation_23 (Activation) activation_24[0][0]	(None, 160, 160, 16) 0	batch_normalization_24[0]
activation_11 (Activation) activation_12[0][0]	(None, 10, 10, 8) 0	batch_normalization_12[0]
concatenate_13 (Concatenate) [0][0]	(None, 160, 160, 32) 0	activation_22
activation_23 [0][0]		activation_23
dropout_3 (Dropout)	(None, 10, 10, 8) 0	activation_11

[0][0]

conv2d_24 (Conv2D) [0][0]	(None, 160, 160, 2)	578	concatenate_13
flatten_1 (Flatten) [0]	(None, 800)	0	dropout_3[0]
batch_normalization_25 (BatchNo [0]	(None, 160, 160, 2)	8	conv2d_24[0]
dense_1 (Dense) [0]	(None, 256)	205056	flatten_1[0]
activation_24 (Activation) activation_25[0][0]	(None, 160, 160, 2)	0	batch_normaliz
dropout_4 (Dropout)	(None, 256)	0	dense_1[0][0]
conv2d_25 (Conv2D) [0][0]	(None, 160, 160, 1)	3	activation_24
dense_2 (Dense) [0]	(None, 1)	257	dropout_4[0]
batch_normalization_26 (BatchNo [0]	(None, 160, 160, 1)	4	conv2d_25[0]
reshape_1 (Reshape)	(None, 1, 1, 1)	0	dense_2[0][0]
activation_25 (Activation) activation_26[0][0]	(None, 160, 160, 1)	0	batch_normaliz
conv2d_transpose_1 (Conv2DTrans [0]	(None, 160, 160, 1)	25601	reshape_1[0]
multiply_1 (Multiply) [0][0]	(None, 160, 160, 1)	0	activation_25
=====	=====	=====	conv2d_transpo
=====	=====	=====	se_1[0][0]
Total params: 2,945,191			=====
Trainable params: 2,916,138			=====

Non-trainable params: 29,053

---



## Direct DL Implementation

```
In [7]: #Assign file list with a list of file to be processed
fileList=demoFiles.copy()

#Get predictions
fileROIs,fileScores=directPredictListWSI(model, xTrainMean, xTrainStd, slicePath,
#Getting the scores and saves the predictions
testingScores=directDLResults(slicePath, fileList, fileROIs, fileScores, annotPat
#Print testing scores
print(scoresDict(testingScores))

49 images saved successfully to AnnotDetected\10039
419 images saved successfully to NonAnnotDetected\10039
468 rows stored successfully to D:\UniData\Projects\NervesSegmentationProject\dataFiles\pred csv\DirectDL\DirectPredFile 10039.csv
23 images saved successfully to AnnotDetected\10097
511 images saved successfully to NonAnnotDetected\10097
534 rows stored successfully to D:\UniData\Projects\NervesSegmentationProject\dataFiles\pred csv\DirectDL\DirectPredFile 10097.csv
15 images saved successfully to AnnotDetected\10114
1059 images saved successfully to NonAnnotDetected\10114
1074 rows stored successfully to D:\UniData\Projects\NervesSegmentationProject\dataFiles\pred csv\DirectDL\DirectPredFile 10114.csv
Testing results :
10039 : Identified annotations [31, 35] and Additional 419
10097 : Identified annotations [18, 19] and Additional 511
10114 : Identified annotations [13, 14] and Additional 1059
```

## Demo

```
In [8]: from demo import demoFile

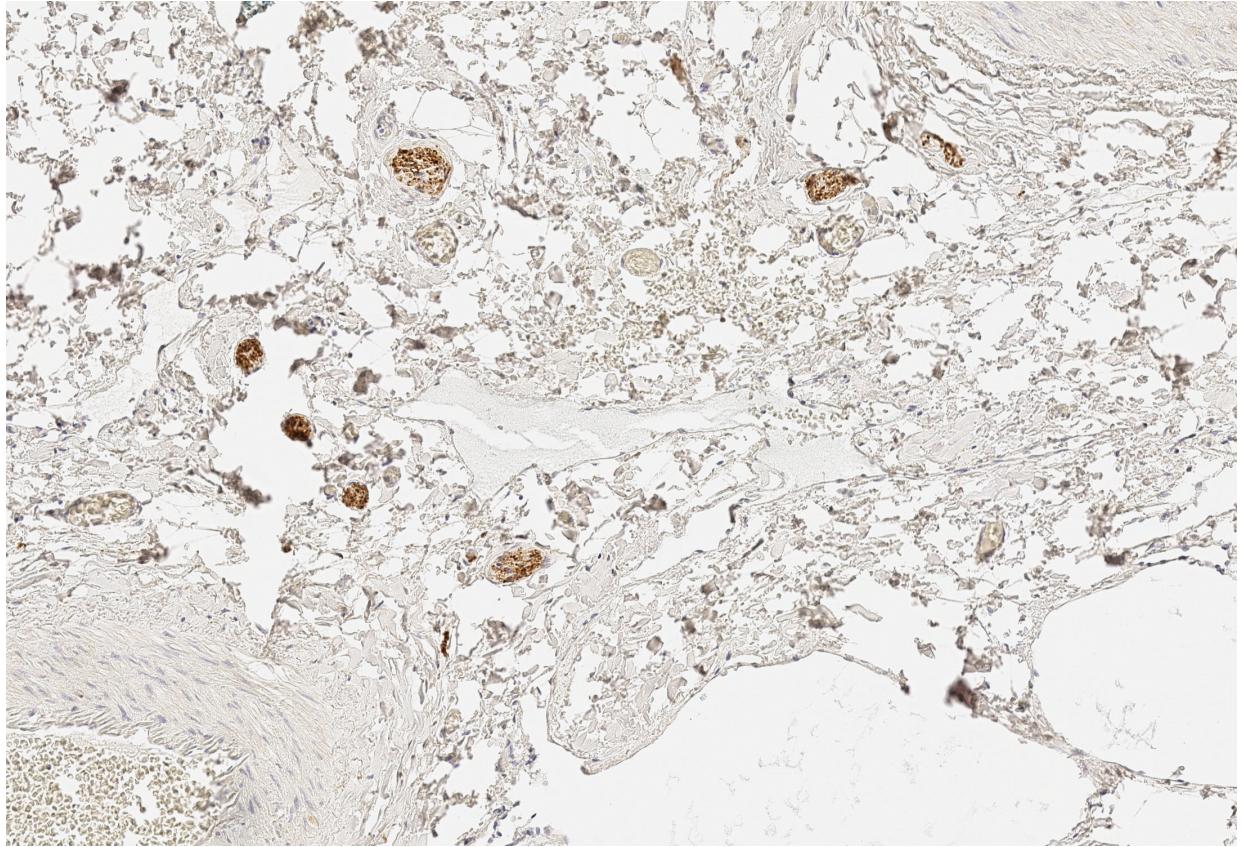
fileList=demoFiles.copy()

#Get the area of the slide to be processed for demo as size of the whole slide is
dArea=getRowsFromCSV(os.path.join(fileDir,"sArea_Demo.csv"),'Dict')
```

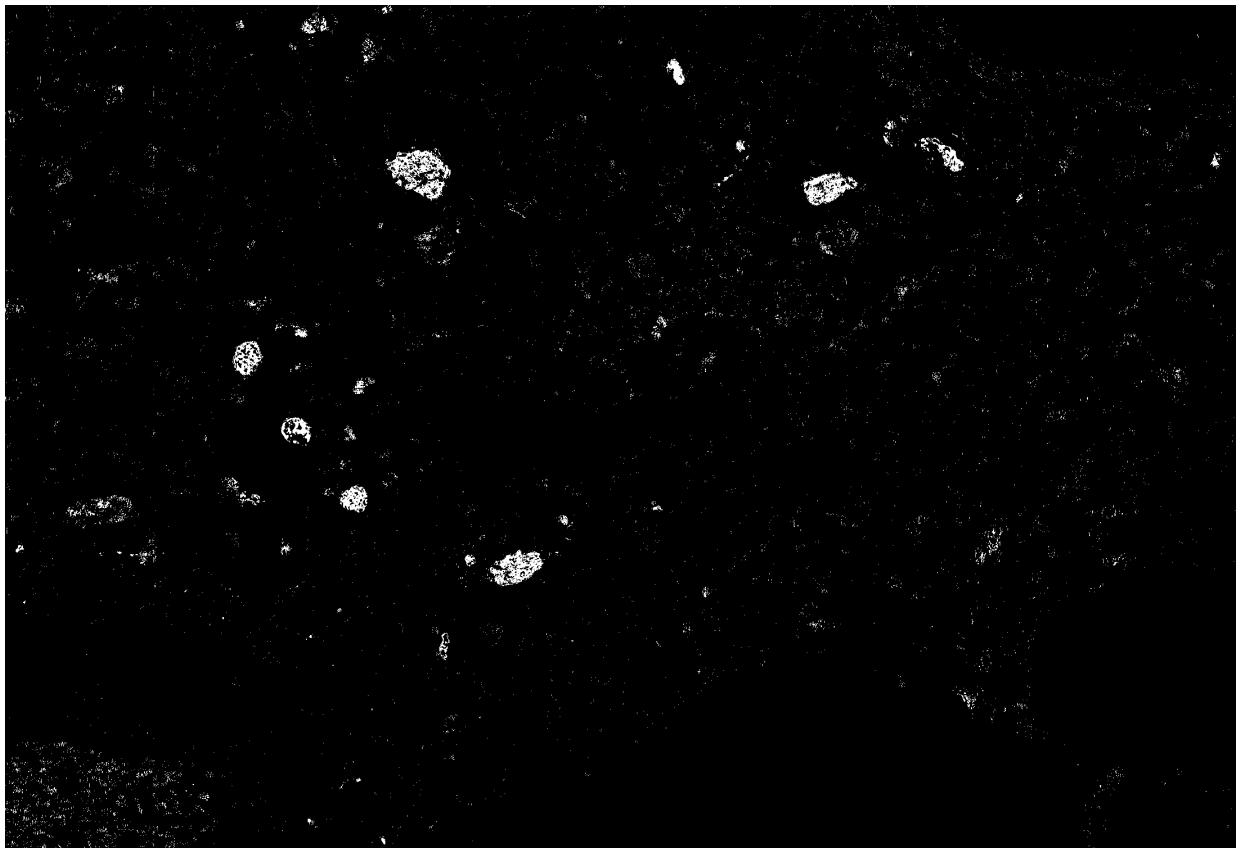
**Draw and visualise prediction on demo file 1 (Scroll for complete results)**

```
In [9]: print("Processing file: "+str(fileList[0]))
demoFile(fileList[0], dArea, fileDir, slicePath, annotPath)
```

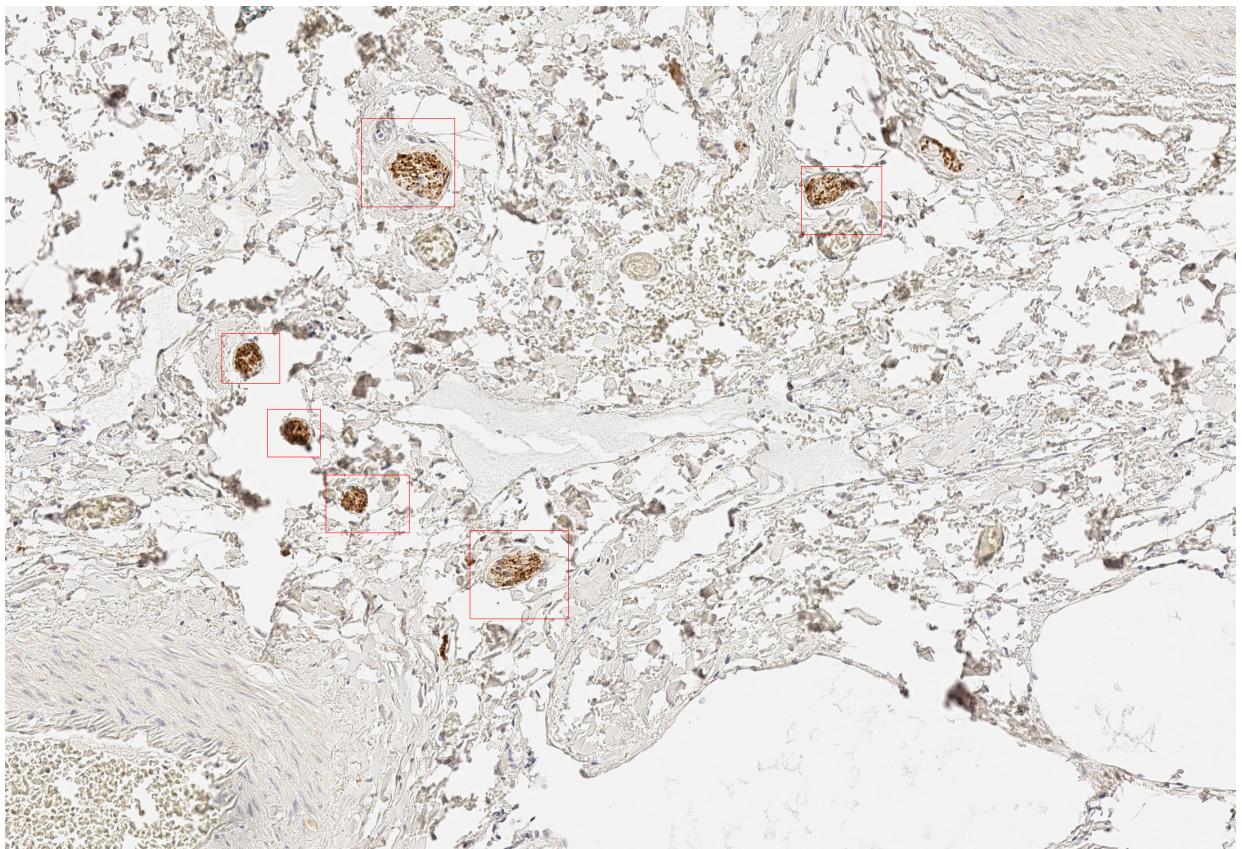
Processing file: 10039  
Drawing original image



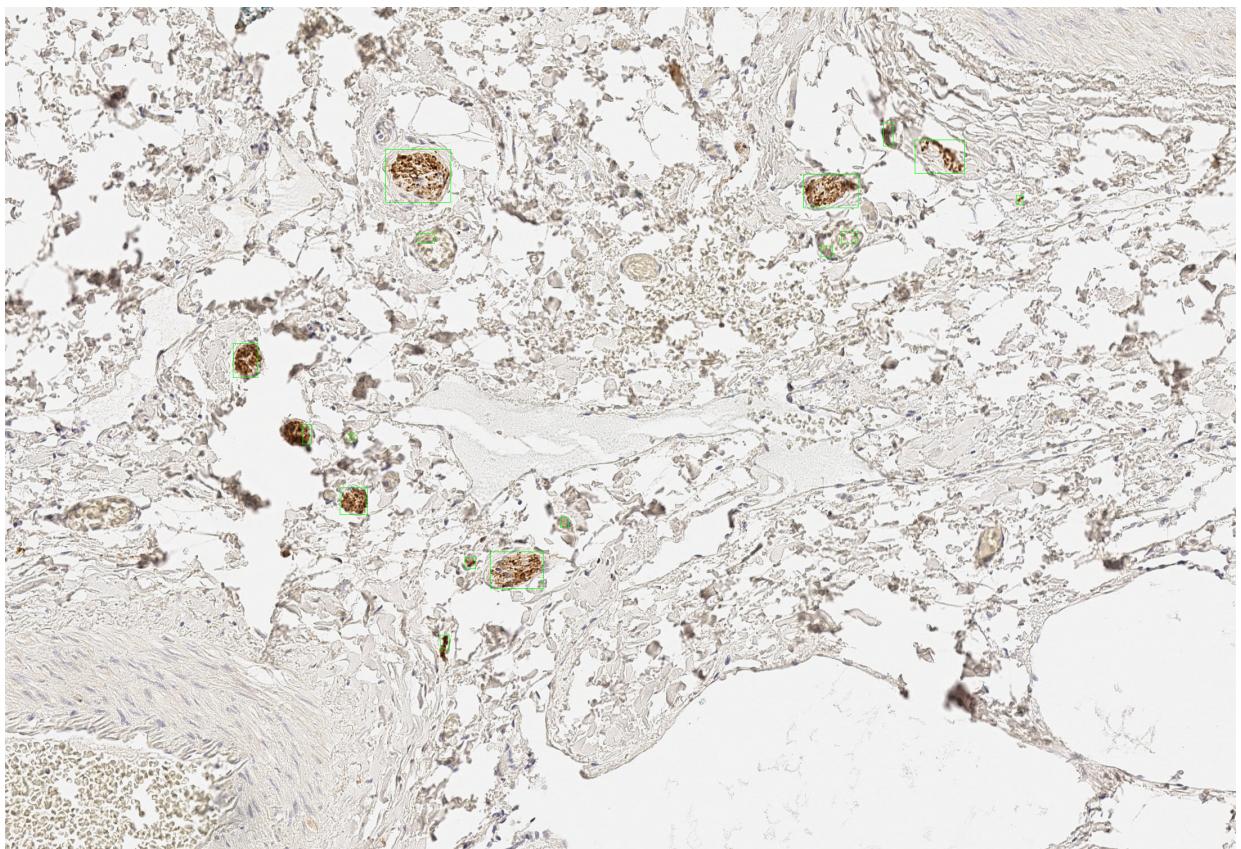
Colour filter output



Drawing manual annotation by experts

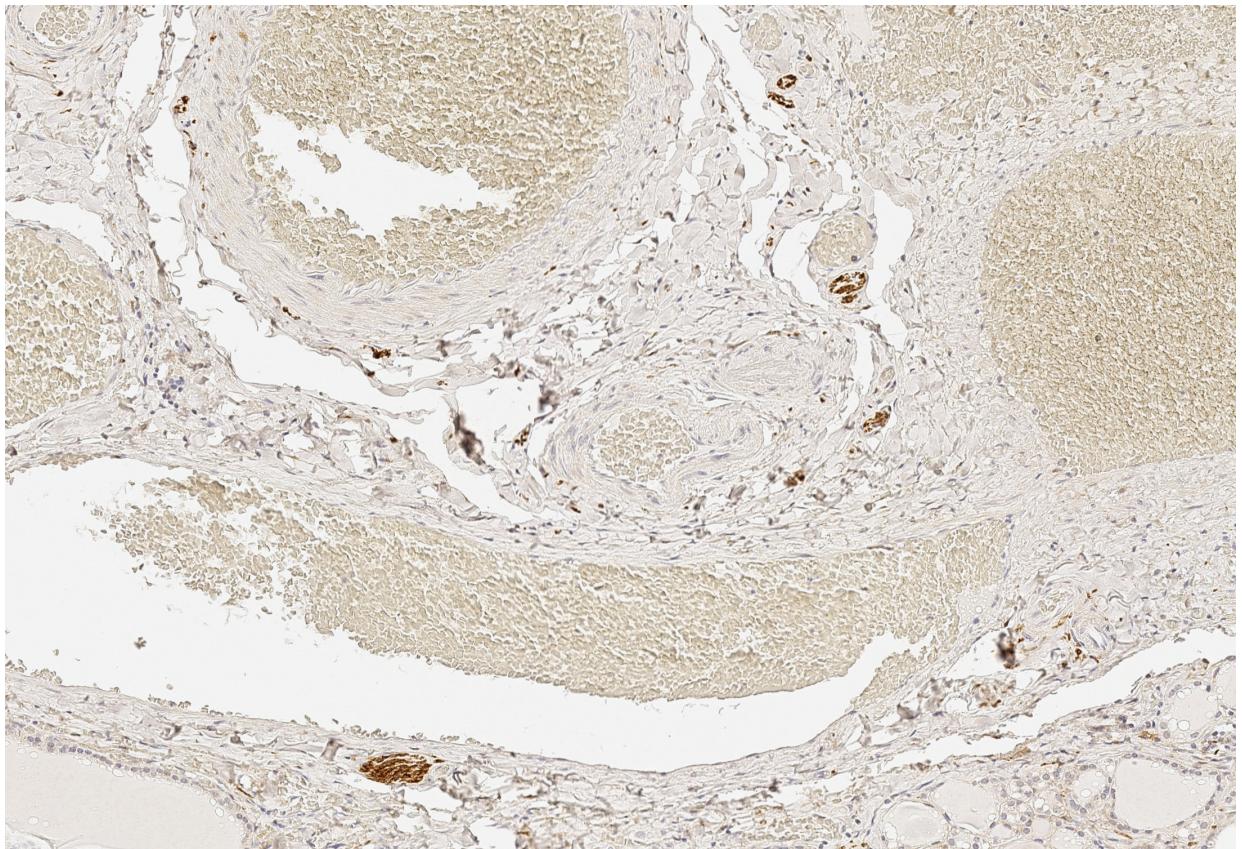


Drawing predictions by model

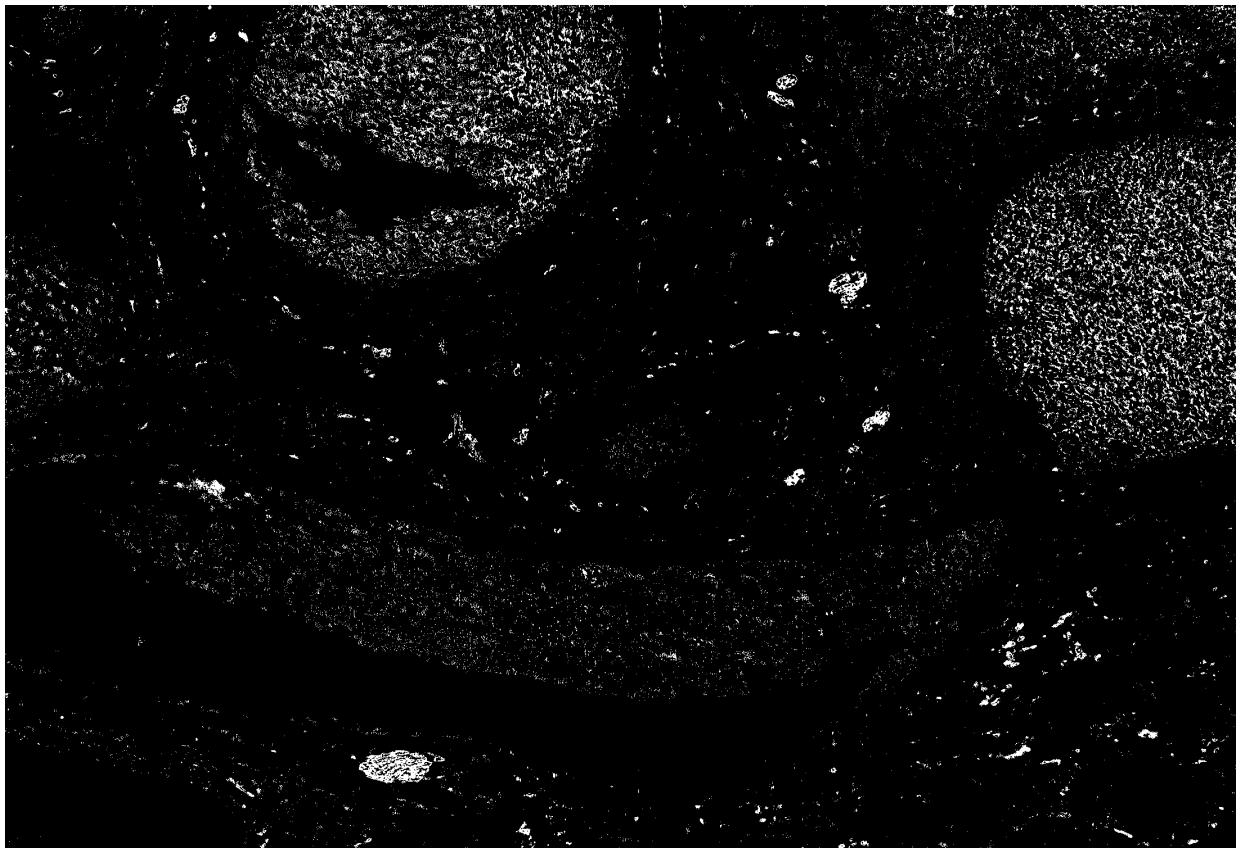


```
In [10]: print("Processing file: "+str(fileList[1]))  
demoFile(fileList[1], dArea, fileDir, slicePath, annotPath)
```

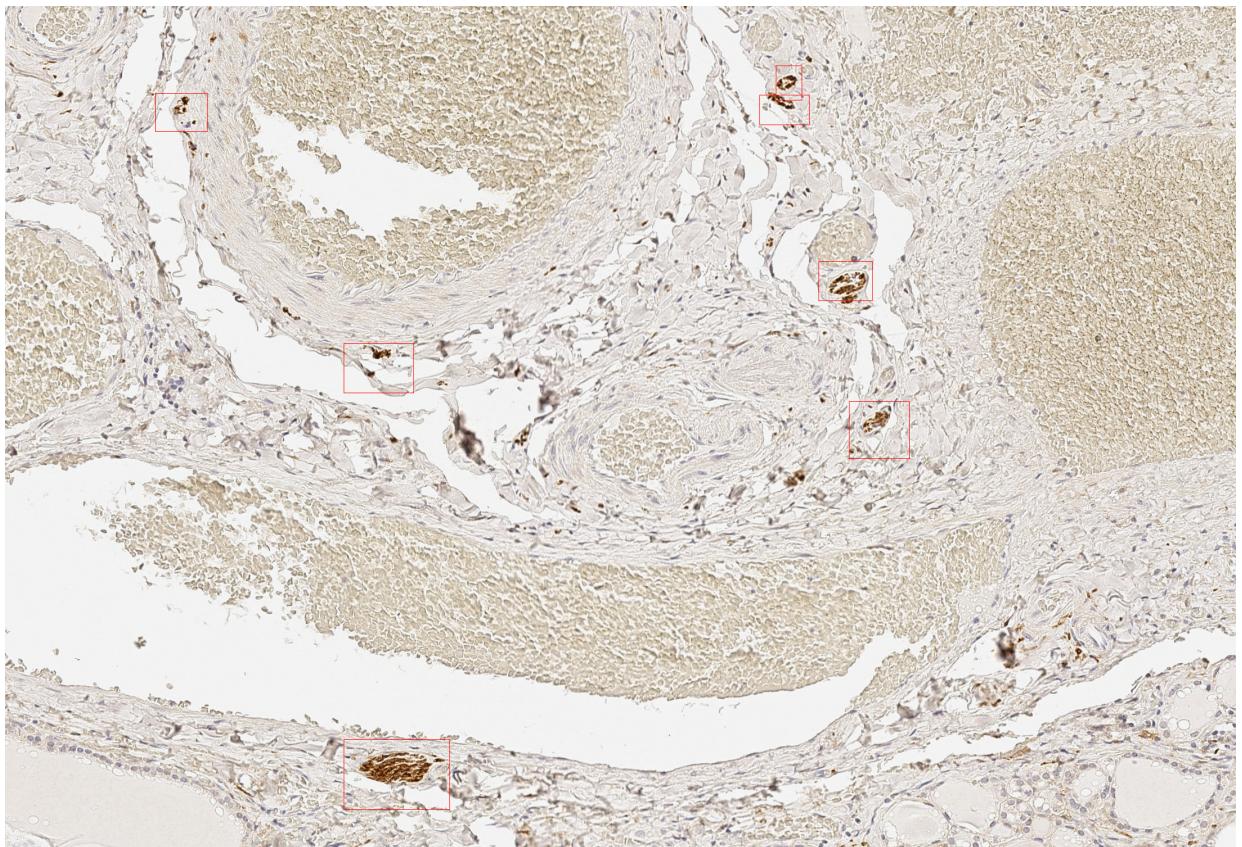
Processing file: 10097  
Drawing original image



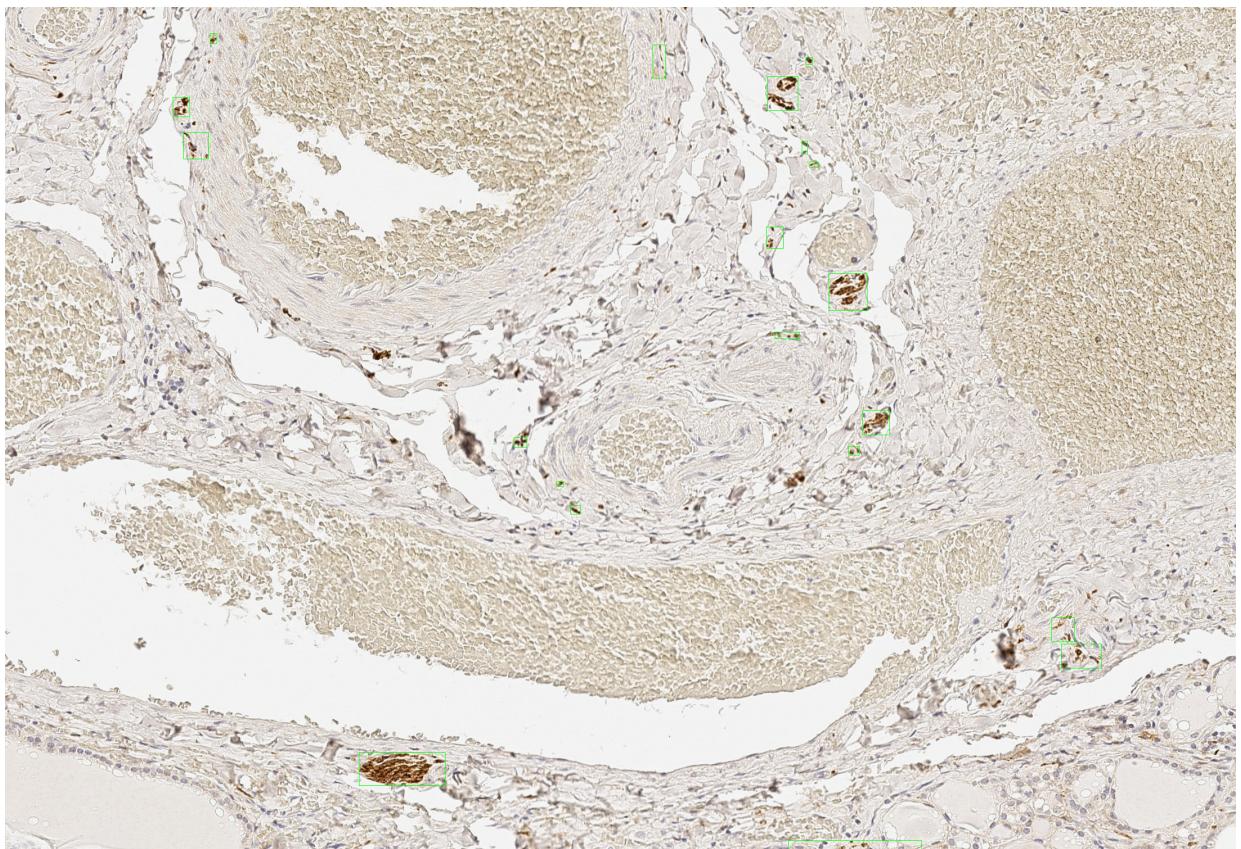
Colour filter output



Drawing manual annotation by experts



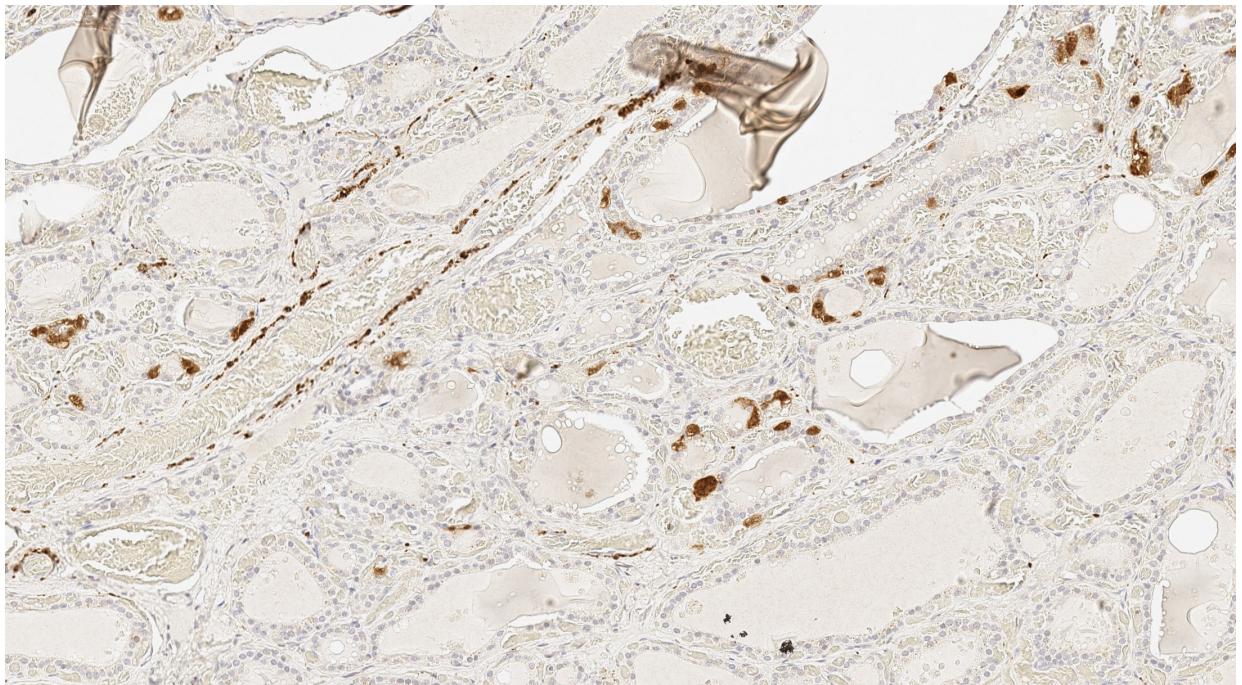
Drawing predictions by model



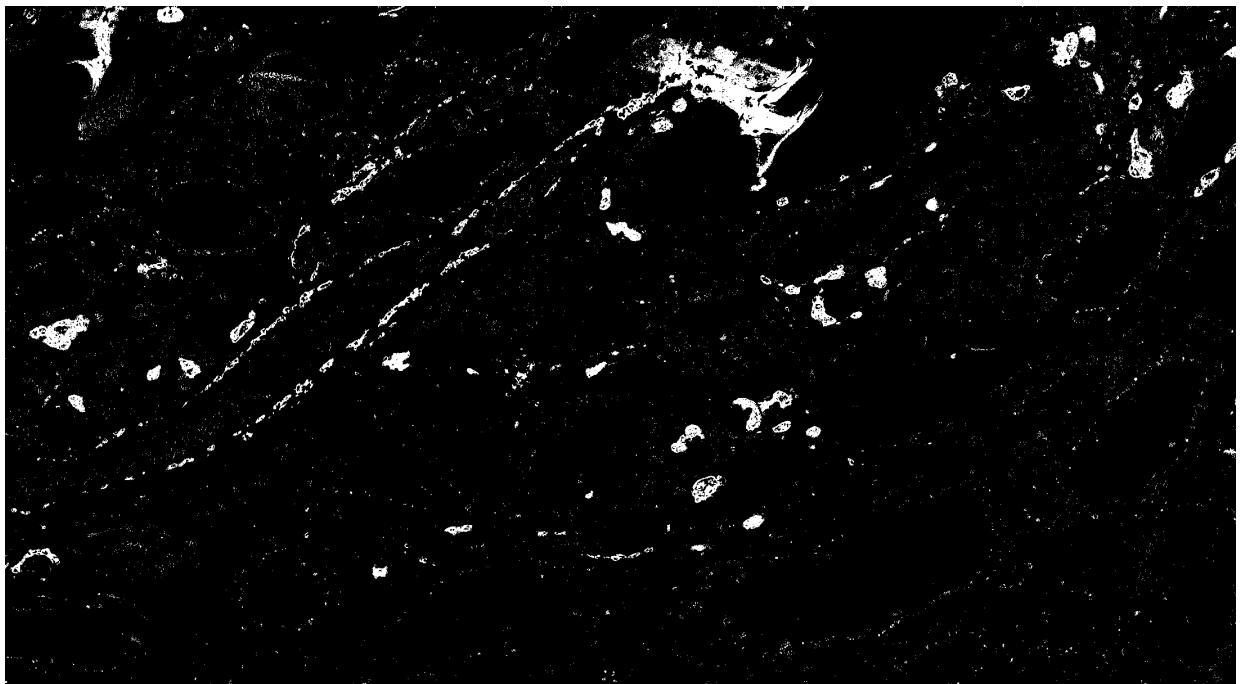
**Draw and visualise prediction on demo file 3 (Scroll for complete results)**

```
In [15]: print("Processing file: "+str(fileList[2]))  
demoFile(fileList[2], dArea, fileDir, slicePath, annotPath)
```

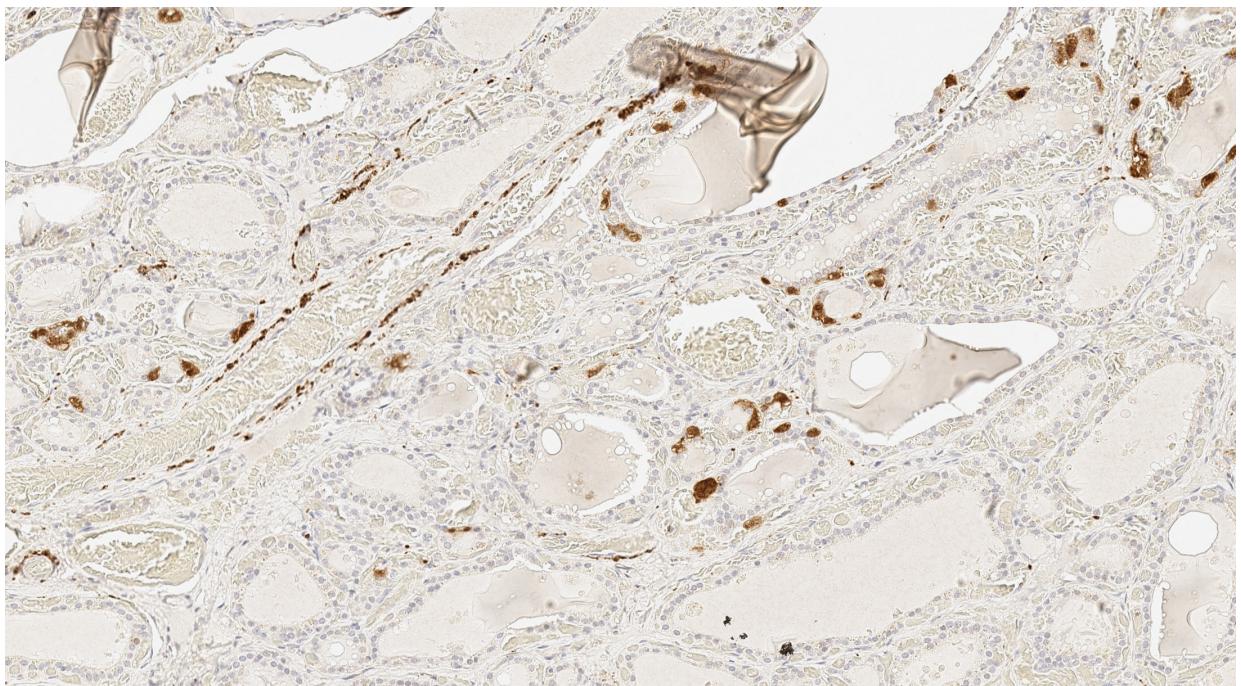
Processing file: 10114  
Drawing original image



Colour filter output



Drawing manual annotation by experts



Drawing predictions by model

