Computer Assisted Classification of Adenocarcinoma Project Summary

Student: Ofer Givoli
Instructor: Dr. Erez Berkovich
Semester: Winter 2013-2014

Contents

1 Introduction

This project extend a previous project [2] done in 2012. The problem domain is images that are the digital scanning of H&E stained tissues taken from the colon. A human pathologist expert segments the images into regions with the one of the following classification: normal tissue, adenoma (almost cancerous) and adenocarcinoma (cancerous).

In the previous project we've dealt with small images that covered very small regions in the slides (each image was in 200x magnification, and was 1280x960 RGB). That projects goal was developing an algorithm that classifies such images into the above classes (normal tissue, adenoma and adenocarcinoma). We've decided on a set of features to extract from each image in order to train and use a classifier.

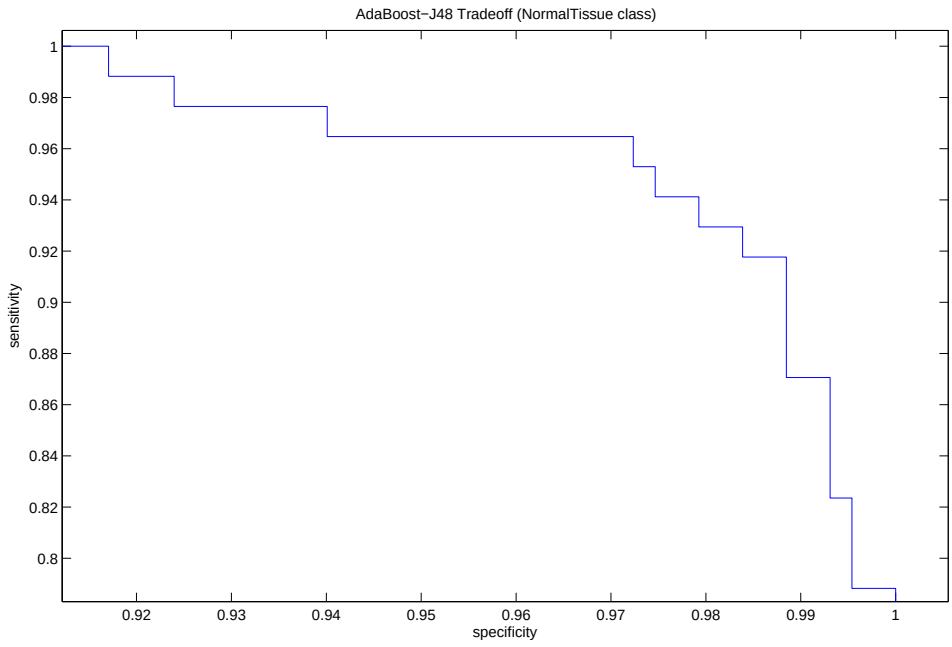
For this project, Dr. Sabo from Rambam kindly allowed me to use his dotSlide digital microscope to acquire large images that cover entire slides. Our goal in this project was to extend our methods from the previous project to

a more real life scenario in which we want to analyze a given full slide and determine whether (and where) it contains adenoma/adenocarcinoma regions.

2 Starting Point

Our starting point for this project was our results from the previous 2012 project. The following ROC curve shows the best performance we managed to get in the previous project, yielded when using the AdaBoost+C4.5 classifier with the following features:

1. Circles_1 - An object-based feature which aspires to count the number of normal crypts in the image - using morphological methods (implemented in matlab).
2. GIST_with_FLD_HematoxylinAndEosinImage - 64 Gabor based numeric features which were extracted (using GIST - a matlab library[1]) from each channel of the HE Image. The HE Image is a representation we invented in the previous project. Each pixel in the HE image holds a pair of values in the range $[0, 1]$ that represents how much each of the two dyes (hematoxylin and eosin) stained the area of that pixel.



[1] http://people.csail.mit.edu/torralba/code/spatialenvelope/

An interesting result we got was that extracting Gabor features from both channels of the HE images yields better results than extracting the Gabor features from the gray images.

So our starting point is that we have a classifier recognizing normal tissue and cancerous (adenoma + adenocarcinoma) sub-images, that is capable of achieving sensitivity of 98.8% and specificity of 87.1%.

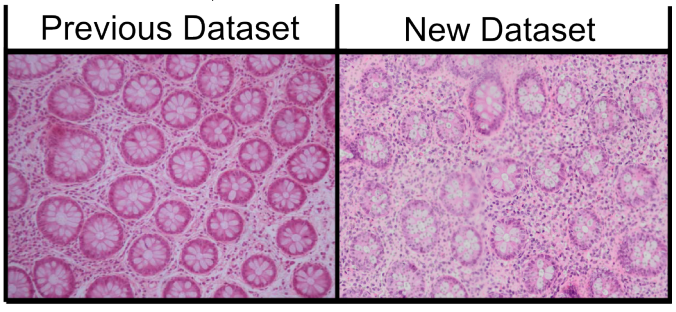
3 Dataset

The acquisition of the new dataset was a challenging task. I've been to Dr. Sabo's office (in Rambam) several times, using his Olympus dotSlide software to extract raw images of full slides that have been scanned using the Olympus dotSlide virtual microscope.

I managed to extract 14 GB of raw image data (after calibrate the system and finding the proper way to extract huge images without memory problems), each image containing a full slide, in batches of 50MB TIFF sub-images. Exporting to one big jpeg2000 was not possible due to memory limitations of the station I've worked on. All in all, the raw image data of 4 full slides were extracted. Two of them are homogeneous (entirely normal tissue, or entirely adenoma). Each of the other two images contained both normal tissue regions and cancerous regions (adenoma/adenocarcinoma), and those images were segmented into those regions. The classification/segmentation was done by Dr. Sabo, and we consider them as ground truth.

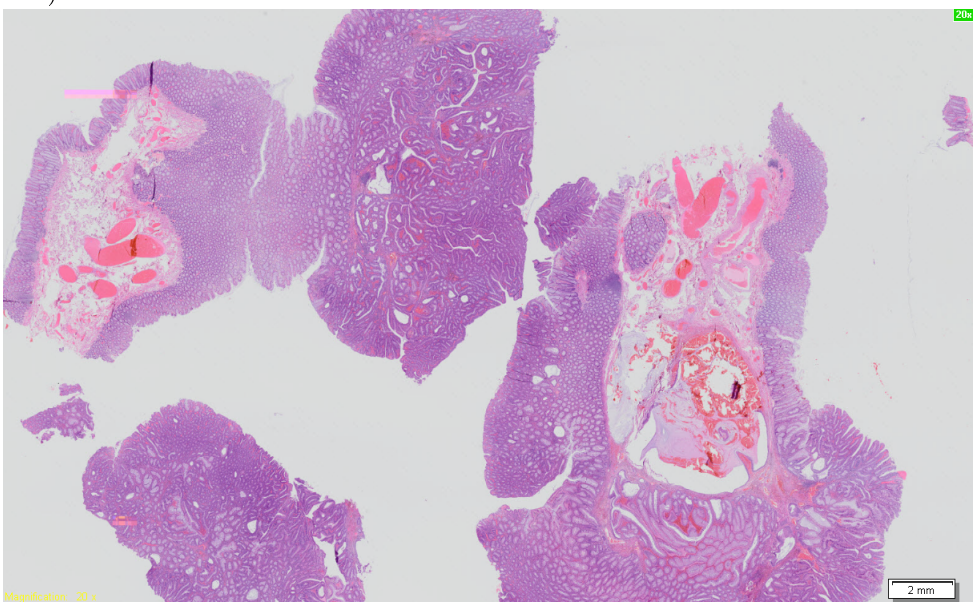
The process in which Dr. Sabo segmented for us those two images was the following: He and I both browsed each large image, and he told me verbally where's the boundary between different class regions, and what's the classification of each region. I've manually logged his input by simply drawing remarks on a very scaled down version of the large images. Later, I've used a matlab GUI tool I wrote to manually mark the boundaries between the class regions - and to export batches of sub-images with a single known ground truth classification.

The following figure demonstrate sub-images from the previous and new datasets (raw images - prior to corrections done on images from the new dataset, described in section 4):



The following image demonstrate an entire full slide scanned (from the new

dataset):



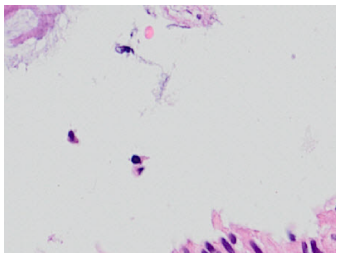
Note: before I realized how problematic the domain was - I've additionally extracted 13 images (15 GB) of slides that we do not have reliable classification/segmentation ground truth for. Someone who's not a pathologist expert (including myself) can sometimes guess the classification/segmentation of those images. I hoped to use these images as a test set (and to have our previous classifier yield a nice sub-images classification, that could be presented as a good segmentation of the full image). Unfortunately, the classifiers I've created during this project were not capable of doing anything useful with those unclassified images.

4 Using our Previous Classifier

The first approach I've tried was using the final classifier from the previous project (trained with sub-images from the previous dataset) to classify sub-images from the current large images in our new dataset. That made sense - as our previous dataset was big (as it was collected over a long period), and I wanted to take advantage of it (preferring at this stage not to train a new classifier using only the new dataset). However, the classifier failed completely (yielding performance comparable to that of randomly guessing the classification). The reason for this failure was that the images on which the classifier was trained on came from a completely different distribution than the test images:

1. First and foremost - the training images are "classical images", in the sense that a human expert selected them (as sub-images from the large images) while trying to aim for "classical normal tissue cases", "classical

adenoma cases" and "classical adenocarcinoma cases". While in the new dataset (served as a test-set here), no one cherry-picked the sub-images. All the sub-images of the large images (that did not contain to much "blank regions") were used. And as a result, most of our new sub-images were not "classical cases" at all. Especially, some of the new sub-images contained a lot of "blank regions" (regions that are completely white - because there was no substance there the dyes could react with). Some of the new sub-images are almost completely "blank". In the previous dataset - images almost never had blank regions. Here's a demonstration of a sub-image from the new dataset containing too much "blank regions":



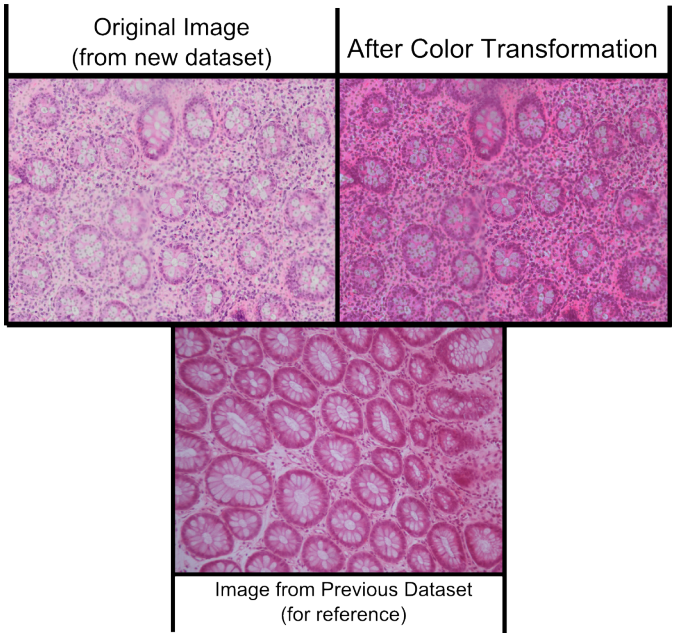
2. The new dataset had a different color distribution than the previous one.

3. The scale of the images in the new and previous datasets were slightly different (but generally - quite similar).

I've tried to overcome those difficulties by doing the following (everything done on matlab):

1. Removing sub-images containing too much blank regions (meaning, too much "white areas" according to a threshold I chose manually). Later I've decided to keep those images and give them a new classification: "None".

2. Performing color transformation on the new dataset, such that the distribution of each of the RGB channels (separately) will match that of the previous dataset. I've implemented this in matlab by combining all the sub-images used from the new dataset into one huge image, and for each channel using the histeq function (with the channel's histogram of the previous dataset I've calculated in advance). The following figure demonstrates the effect of the transformation:



3. Carefully assessing the ratio between the scale of the new and previous sub-images (unfortunately, this had to be done manually by finding the median diameter of normal crypts[2]), and correcting the small scale gap my resizing the new sub-images (via bicubic interpolation).

Even after all these improvements, the performance of the previous classifier was still completely useless (comparable with randomly guessing the classifications).

The classification itself was done with Weka (a machine learning java library), and the feature extraction was done in matlab. I've wrote a C++ program (using Qt) to translate the output features files from matlab to Weka's format (ARFF).

5 Employing Convolution Neural Networks for Classification

Convolution Neural Networks (CNN) are deep neural network classifiers in which some of the hidden layers perform convolution with kernels that are learned. CNNs became very popular in recent years and are used in state-of-the-art implementations in a few domains, especially in voice recognition and image classification. Further explanations on CNN are available here: [1]

[2] I've done this with a matlab GUI I wrote.

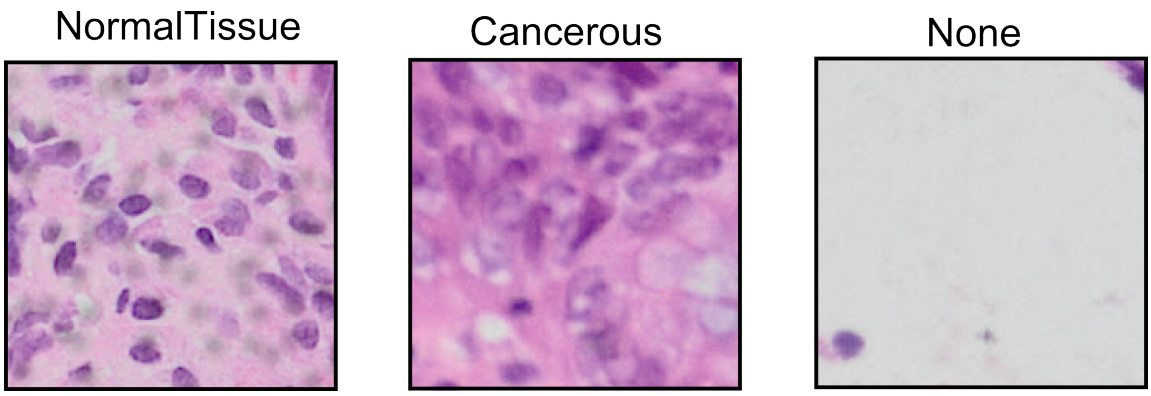
I tried to employ CNNs to our project. The CNN topologies I tried contained 2-3 convolution layers and received 3 channels as input (meaning: the first layer's input was 3 feature maps - each is one of the RGB channels). The process of training a CNN is done iterative. In each iteration a batch of training images is used to modify the weights of the CNN. A sequence of iterations in which all the training images are used once is called an epoch. After finishing an epoch - we start another, and so on. We should stop the process when the error rate no longer decreases.

To train and run the CNNs I've used a package called cuda-convnet (a fast implementation using NVidia's GPUs).

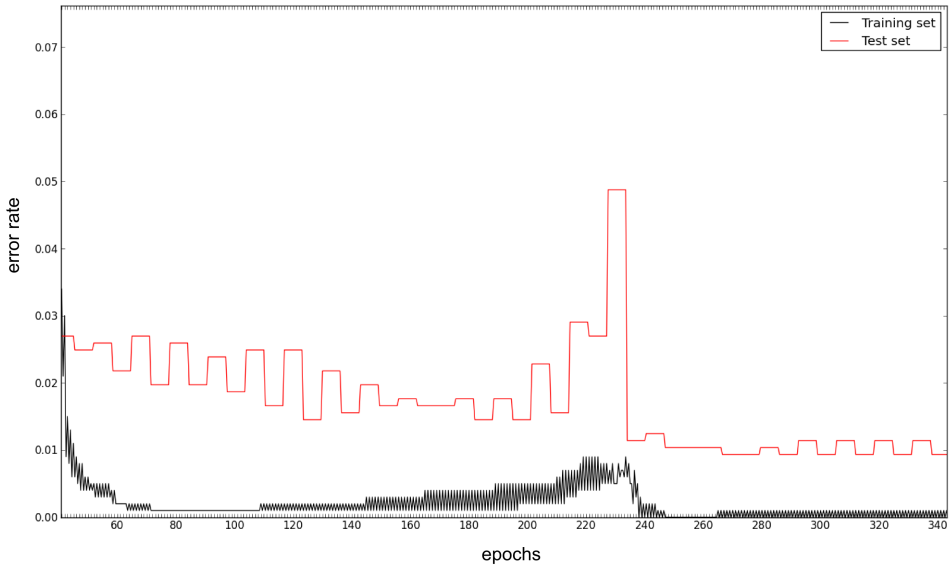
I'll now describe the experiments done. Each experiment was repeated twice: once with 64x64 input RGB images, and once with 200x200 input RGB images (similar sizes are often used as input for CNNs, for image classification problems). Of course - each time with a matching CNN topology. The results were similar, so I'll only present here the results of the 200x200 experiments. Preparing the training/test sets for cuda-convnet was done in matlab.

The first experiment I've done was the following. I've only used subimages from two homogenous slides: one containing only normal tissue and one containing only adenoma. In order to handle subimages containing "blank regions", I've changed the classification of all images that were "too white" to "None". I've defined "too white" as follows: I manually chose a threshold defining the minimum value over the minimum channel of each pixel for it to be considered a "white pixel". Another threshold determined the minimum ratio of "white pixels" for the entire sub-image to be considered "too white").

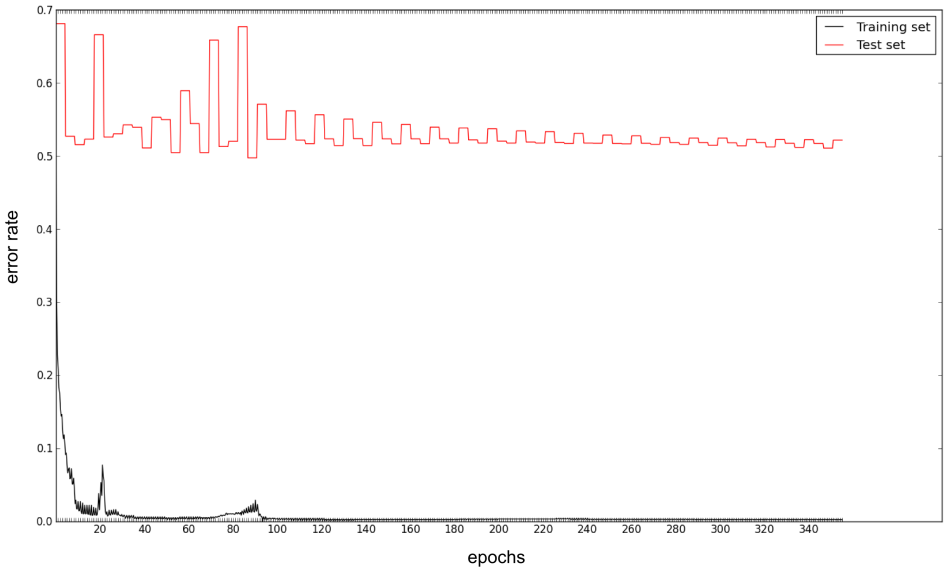
So there were 3 classifications: NormalTissue, Cancerouse and None. I've used 1,204 NormalTissue images, 1,236 Cancerous images and 524 "None" images. Here's an example for each class:



I've used 2/3 of the data for training, and 1/3 as a test set (I've excluded None images from the test set). The first result I got was misleadingly great:



After getting this I was thrilled (1% error rate!). Unfortunately, the next experiment proved I was happy too soon. This time, I took the training set from the same slides as before, but the test set was now taken from a different slide - a heterogeneous one (with both normal tissue regions and adenoma regions). The 3 classifications remained the same: NormalTissue, Cancerouse and None. The test set contained: 1,671 NormalTissue images and 1,690 Cancerous images). This time the results were horrible:



So the classifier didn't really manage to generalize the concepts of NormalTissue and Cancerouse beyond the slide it was trained with. The great results of the first experiment are only attributed to the fact that for each class - the test images were taken from the same slide as the training images.

6 Conclusion

The new dataset collected for this project was too different from our previous dataset, rendering our previous classifier useless for the task of classifying sub-images from the new dataset. Unfortunately, it seems the new dataset is not large enough to allow training useful classifiers that generalize the concepts of normal and cancerous tissues (for any slide), or at least it is so for the learners I've tried. I believe that if a larger dataset was used - CNN might have proved to be a good classifier for this problem domain, and furthermore, the HE image representation (which we've invented in the previous project) might have proved very useful, as suggested from the results of our previous project.

Let us note an additional important conclusion: sub-images from the same slide tend to be similar to each other, and thus the sub-images used to train a classifier must be taken from many different slides. Also, one should not measure the performance of a classifier with test sub-images that are from the same slides as other sub-images used to train the classifier.

References

[1] David Bouchain. Character recognition using convolutional neural networks. *Institute for Neural Information Processing*, 2007, 2006.

[2] Ofer Givoli. Computer assisted classification of adenocarcinoma. *Center for Intelligent Systems, Technion - undergraduate project*, 2012. Instructed by Dr. Erez Berkovich. Research summary available at: [http://goo.gl/cb0pWk].