

Prostate Tumor Segmentation

CS188 Spring17

Brian Li (504442822)

Daniel O'Laughlin (304467494)

The purpose of our project is to develop a MATLAB program that implements automatic segmentation of a prostate tumor using supervised machine learning. Our project draws from a dataset of MRI images obtained from our professor, Dr. Fabien Scalzo. Each MRI image of a tumorous prostate is also paired with a mask, which is an image of the sample in which the tumor and prostate are already segmented. MRI Images that did not have obvious tumor correlation with the tumor mask were filtered out by the authors of this project, leaving seven remaining images as the dataset.

METHOD

This section gives an overview of the complete process of the implementation of automatic segmentation. First, the given masks in the dataset are employed. Each mask represents the prostate tumor with yellow pixels, the non-tumorous parts of the prostate with teal, and the non-prostate (background) parts of the image with blue. Instead of using the RGB values, each segment of the mask also has an index value, with '0' representing the background, '1' representing the non-tumorous parts of prostate, and '2' representing the tumor. By looping through the pixel array of the sample and checking the index value of each pixel, three new separate masks can be created. The (x,y) coordinates of the tumor, healthy prostate parts, and the background are also stored in a tumor_array, healthy_array, and background_array respectively.

The next step is to create random patches of tumor, non-tumorous, and background pixels. 100 randomly chosen pixel coordinate values are taken from each array, and patches with different diameters (1, 3, 5, and 7) are created at the 100 pixels. The intensity of the pixels in each patch is stored into a vector. For example, with patch diameter 3, the patch will have 9 intensity values stored in a row. There are 100 columns in the vector, corresponding to the 100 random pixels. This vector will serve as the input data for the supervised regression model used later on.

An array of 100 values is also created, filled with either values of either all '0's, '1's, or '2's. If the input data was obtained from healthy_array, the array would be filled with one hundred '0' values,

'1's for tumor_array, and '2' for background_array. This array serves as the response values for the supervised regression model. Thus, for a single sample image, with patch diameter of 3, three different 9x100 vectors and three different 1x100 arrays will be created. The three vectors and three arrays are then concatenated together. This process is repeated for the seven different images in the dataset and the resulting vectors and arrays are also concatenated together to create one large $d^2 \times 2100$ vector and one large 1x2100 array, where d is the diameter.

The $d^2 \times 2100$ vector, which serves as the input data, and the 1x2100 array, which serves as the response values, are then used in supervised machine learning. The MATLAB Regression Learner app is used to train regression models with algorithms such as Linear Regression, Generalized Linear Models, Nonlinear Regression, Support Vector Machine Regression, Gaussian Process Regression, Regression Trees, and Model Building and Assessment. For the purposes of our project, the Gaussian Process Regression algorithm was chosen to train our regression model because it resulted in the lowest Root Means Square Error (*RMSE*), which measures the difference between the value predicted by a model and the actual observed value.

After applying the Gaussian Process Regression algorithm to train our model, the resulting model is exported to MATLAB. This model can now be used to predict response values for a new input image. The model takes in the array of patches from the new input image. Every pixel will be included the array so that the model can take in the intensity of every pixel and calculate response values. These values, which range between 0 and 1, will be either rounded up or down depending if it is greater or less than 0.5. The response values of every pixel, either 0 or 1, now can be used to generate an image with 2 segments, either tumorous or non-tumorous prostate. The process is now finished and can be repeated on any number of MRI images to segment prostate tumors.

EVALUATION

The authors of this project believe that this project's method is an effective approach to prostate tumor segmentation, based on the results in the following section. However, it is important to analyze and evaluate the benefits of the project method as well the

ways it could improve in order to obtain an exhaustive understanding of the results. Firstly, accuracy of the project's findings was increased by using differently-numbered pixel patches, instead of just each pixel intensity value. However, the project could greatly benefit from more data. While the original data set contained 64 samples, most samples did not have visible dark spots at the locations noted by the sample's mask.

RESULTS

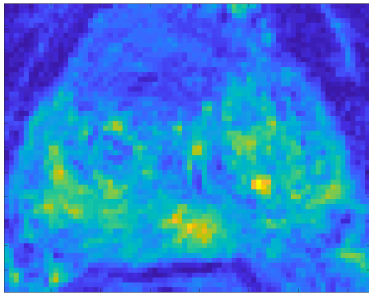


Figure 1: MRI Snippet of Prostate

Figure 1. shows an MRI snippet that contains just the prostate. Figures 2. and 3. show the ground truth and the binary map generated by our machine learning algorithm respectively. Note that the tumor in Figure 2. is displayed in yellow. Also note our algorithm was run on Figure 1 and since the prostate is the majority of the MRI Snippet image, our results do not segment the prostate itself. Rather we simply attempt to segment the position of the tumor within the image.



Figure 2: Mask Showing Actual Location of Tumor

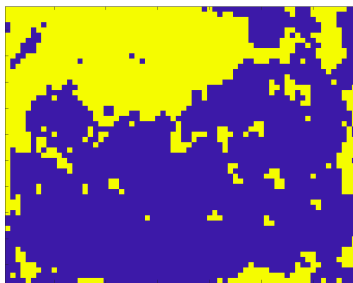


Figure 3: Mask Created via our machine learning program

Another example of an MRI snippet of prostate, a tumor mask, and our created mask can be seen on the supplementary figures Page (Figures 5,6,7).

As you can see from the above examples, our masks do tend to cluster pixels at the area of the tumor, however variable amounts of pixels are also misclassified as having been part of the tumor. This is likely a product of our extremely small data set. The results above were generated using a learned model generated from data taken from only 4 images. However, when the model was trained with more data, the results were better. The example from Figure 1

was done using a model that was trained on 7 images (this was the max amount of decent MRI images we had) and the results were better:

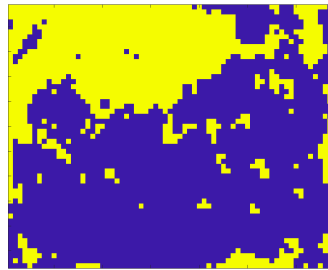


Figure 3 (again): Tumor Segmentation of Figure 1. Done by a model trained from 4 MRI's

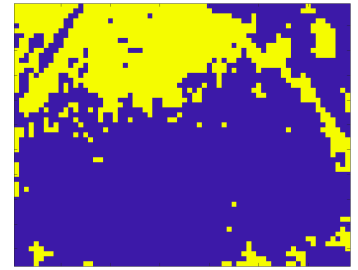
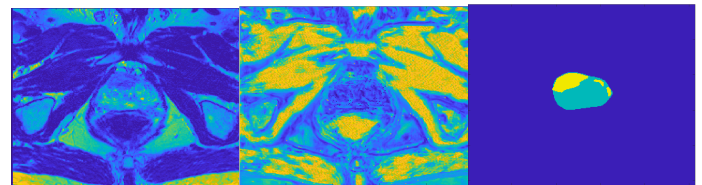


Figure 4: Tumor segmentation of Figure 1 done by a model trained from 7 MRIs

There is misclassification in the image (Figure 4) generated from a more "intelligent" model. The segmenting of the entire MRI images (not just prostate snippets) proved to be a bit more difficult because our training data had to not only include patches of healthy prostate and tumorous prostate, but also patches of "background" MRI data. A heat map generated from our machine learning algorithm is shown below.



From the segmented MRI heat map obtained from our machine learning algorithm it is not readily obvious where the tumor resides. A possible explanation for this relative vagueness is as follows:

since training patches contain pixel intensity values, similar intensity values outside of the prostate coupled with very sparse training data makes it extremely difficult to describe a clear separation between the prostate tumor and the areas of the MRI image.

DISCUSSION

With the increased usage of MRI for the purposes of clinical study of prostate cancer, segmentation in this area can improve efficiency and be useful for determining boundaries for radiotherapy [1]. Our project set out to develop prostate tumor segmentation using supervised machine learning.

Through testing different patch diameters, it was also found that 7x7 patches results in an increasingly decreasing RMSE value, compared to 5x5 patches and 3x3 patches (See Figure 1, Figure 2, and Figure 3 on the following page). As stated earlier, with more training data, the Gaussian Process Regression created a more model which predicts a more accurate tumor segmentation.

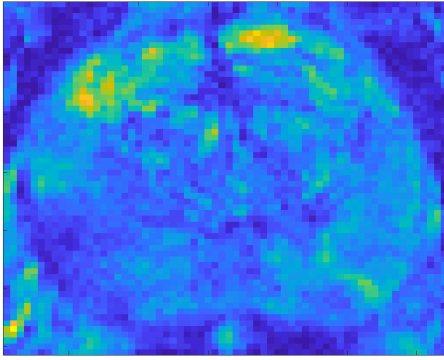


Figure 5: MRI Snippet of Prostate



Figure 6: Mask Showing Actual Location of Tumor

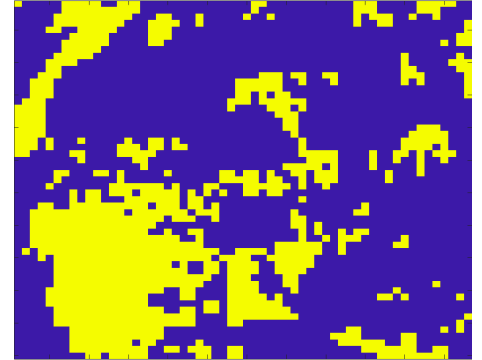


Figure 7: Mask Created via our machine learning program

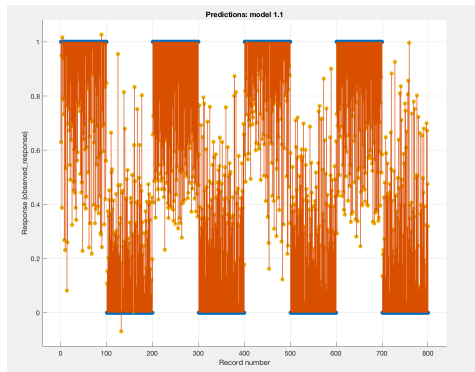


Figure 8. Gaussian Process Model obtained by using 5x5 patches

Results
 RMSE 0.45
 R-Squared 0.19
 MSE 0.20
 MAE 0.41

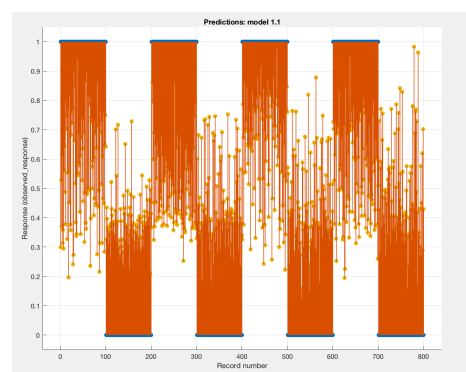


Figure 9. Gaussian Process Model obtained by using 3x3 patches

Results
 RMSE 0.47
 R-Squared 0.11
 MSE 0.22
 MAE 0.44

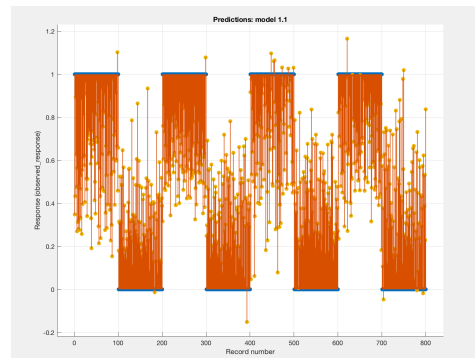


Figure 10. Gaussian Process Model obtained by using 7x7 patches

Results
 RMSE 0.43
 R-Squared 0.26
 MSE 0.19
 MAE 0.38

Sources

[1]. Litjens, Geert et al. "Evaluation of Prostate Segmentation Algorithms for MRI: The PROMISE12 Challenge." Medical image analysis 18.2 (2014): 359–373. PMC. Web. 10 June 2017.