

Eigenface Recognition as it Pertains to Medical Diagnosis

Final Report

Chad Newton

University of Michigan: Biomedical Engineering Department
EECS 451
Ann Arbor, MI
newtonca@umich.edu

Abstract—This paper articulates the benefit of using an Eigenface algorithm to sort breast MR images from other MR images such as an image of brain or lower abdominal cavity. This is the first step to make a program that will detect breast lumps from MRI's and decrease the time needed to get a medical diagnosis back to a potential breast cancer patient.

Keywords—breast cancer; Eigenface; magnetic resonance

1) Introduction

A smart phone these days will allow you to secure your phone by simply showing your face. This facial recognition software, at least some, are based upon the concept of Eigenfaces, which are ultimately based on the mathematical eigenvector. Simply put the program uses this algorithm to measure the difference between a known image (the face one would use to set up the smartphone lock) and an input image (the face the phone detects after the lock has been set up). What I am attempting to accomplish is to develop a method that could set computerized medical diagnosis just one step further. In this report are the beginning steps of implementing an Eigenface algorithm to take a known set of breast lump MR images and measure the difference between the known set and a new input signal to automatically detect whether or not the new signal belongs to the known set.

2) Data Set

The data for this project came due to the Cancer Imaging Archive (CIA). This website allows researchers to have open source access to any sort of medical image they have on their database. For my purposes, I selected three patients that I deemed to have lumps in their breast, one patient that I found no lumps in to use as a control, and a few random MR images of regions not of the breasts.

From here, I would assign the three first patients the designation of s1, s2, and s3 for sample 1, sample 2, and sample 3 respectively. The control I labeled n3, and the random images I loaded into the system when the time was needed and did not assign any particular variable in Matlab.

3) How the Algorithm Works

The goal of this project is to create a reference set from the samples we get from the CIA and measure the difference between this set and a test image

- We get the reference set by designing a struct set in Matlab that contains all images that we want in it. Then we take the mean of the struct set and subtract its mean from itself to normalize whole the set.
- To measure the difference between the reference set and the new image, we have to form an Eigenbasis for the reference set. An Eigenbasis is a basis from which all Eigenvectors can be formed. We do this by taking the normalized struct set and forming a matrix A. From here, we make the covariant matrix C by multiplying the inverse of A by A. We get the Eigenbasis from matrix C
- From the Eigenbasis weights are assigned to matrix A and the test image. The difference between these weights determines the correlation between the two. The smaller the distance, the larger the correlation.

Figure 1 depicts this process.

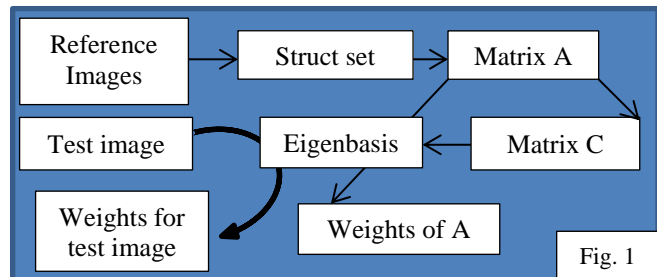
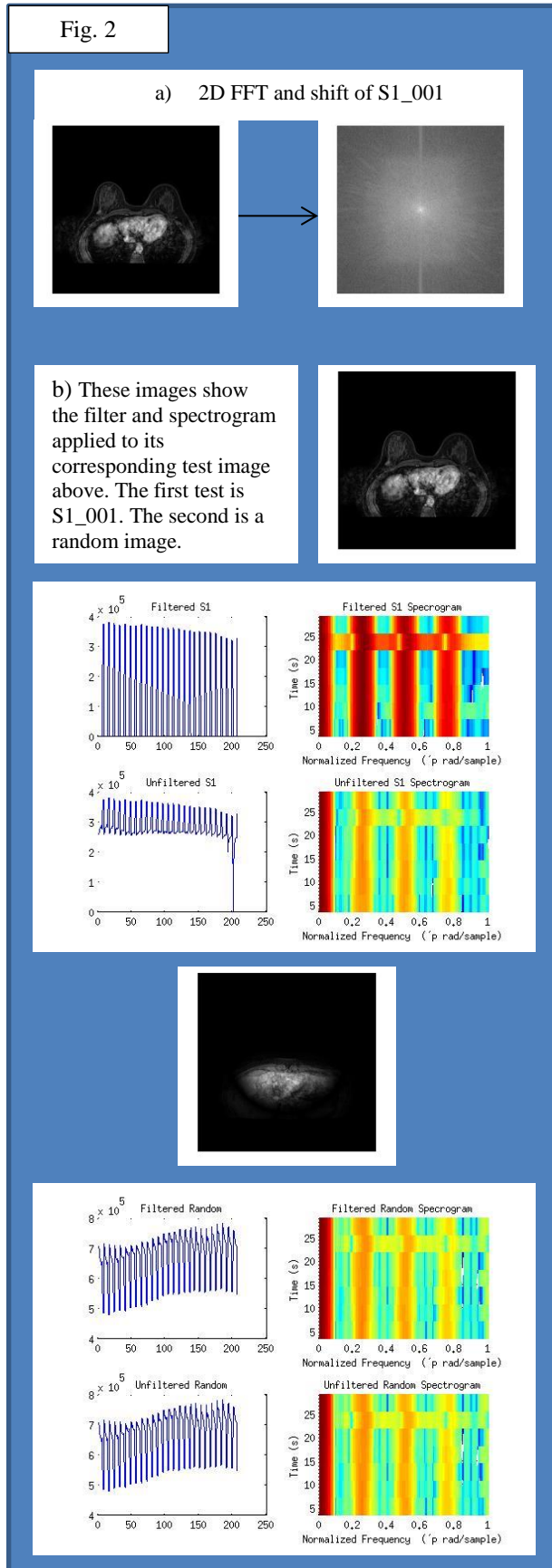


Fig. 2



4) In-Class Tools Used

There are three tools of which we learned this semester in class that this algorithm uses:

a) The 2D FFT

This function transforms the time domain MR images in to the Fourier domain. This allows the algorithm to detect the differences between the reference set and the test image. When I tried to run the algorithm in the time domain, it could make no real distinction from one image to another. The code took the FFT and carried out the FFT-shift to eliminate the complex numbers from the initial transform.

b) Filtering

During the preliminary tests of the efficacy of the algorithm, I found that images that had a high correlation had a weight coefficient of about 300,000. This filter emphasizes a trait that already somewhat exists. To accentuate the differences between images that were alike and those that were not, all weight coefficients below 300,000 were then valued to 0 so the final average correlation would be lower for the images that belonged in the same group.

c) Spectrography

The algorithm uses the spectrogram function to give the program a more visual side. Since it primarily relies on giving a number to determine an image's correlation to the reference set, casting a spectrogram of the signal gives the user a better understanding of the correlation without having to use numbers at all. This tool combined with the filter listed out in B gives the user the best ability to determine an image's correlation.

These tools can be seen in Figure 2

5) Out-of- Class Tools Used

a) Resizing an Image

For the algorithm to work, all of the images used must have the same dimensions. Unfortunately when images are collected, they are not always the same size. This is resolved by the 'imresize' function in Matlab. For this project the dimensions were set to 640x640.

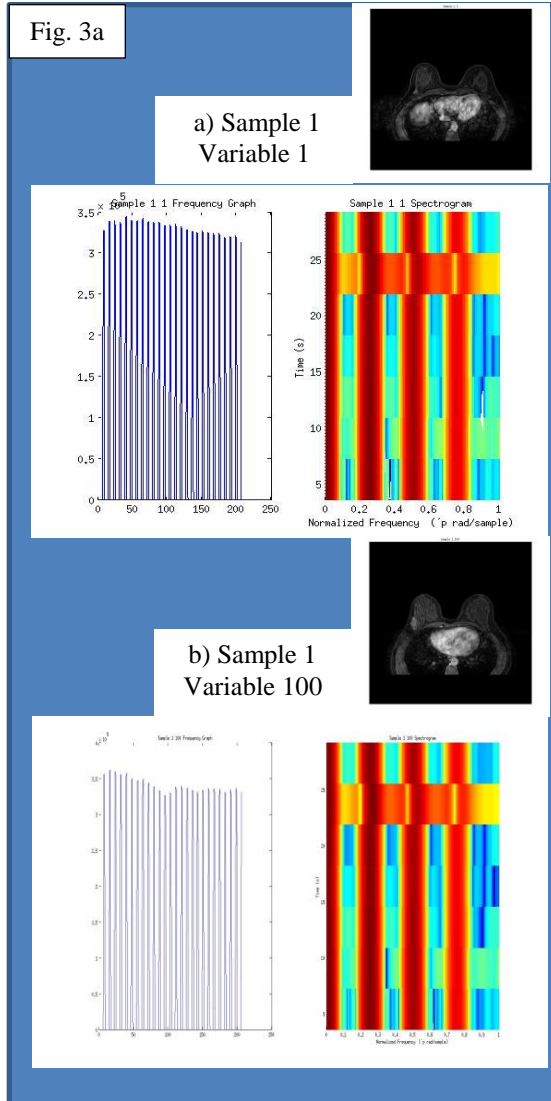
b) Eigenvalues and Eigenvectors

As described in 3, matrix C defines the Eigenbasis from which weight values are assigned to the test image and matrix A. We get this Eigenbasis from the 'eig' command that creates the Eigenvalues needed from C.

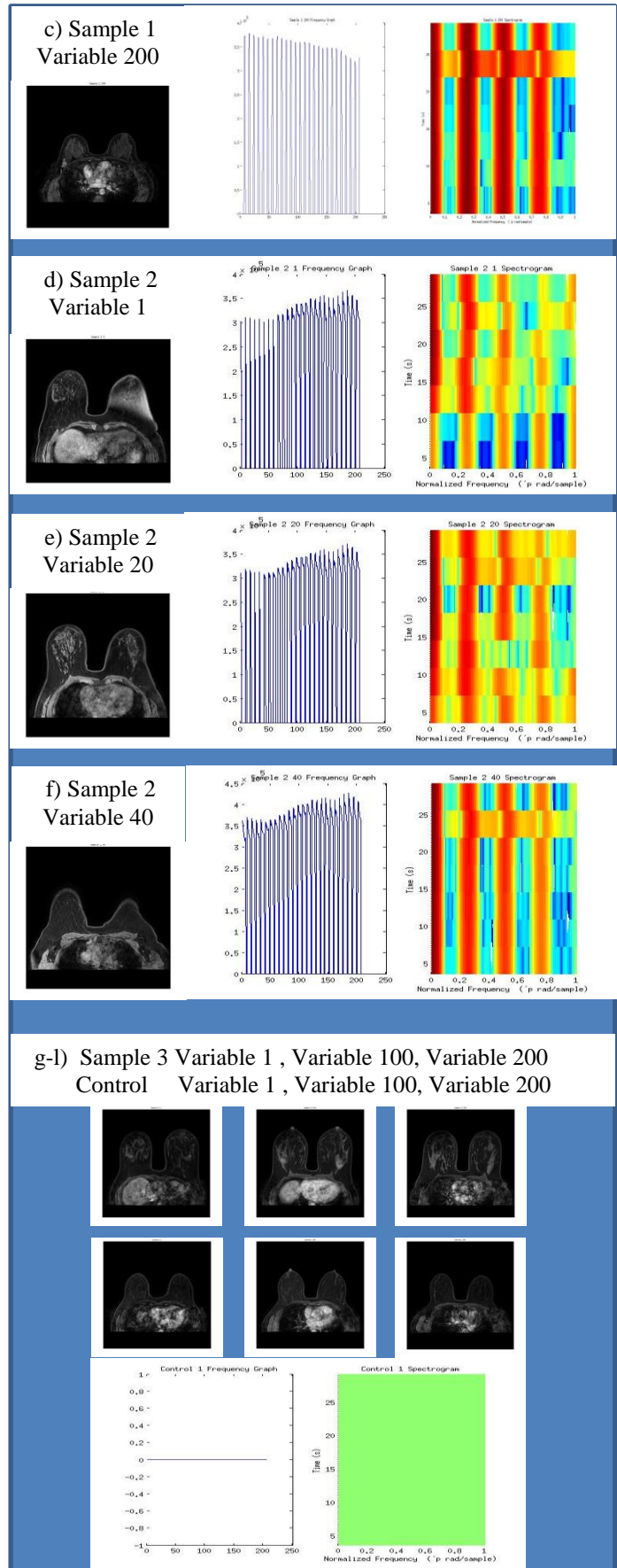
6) Results

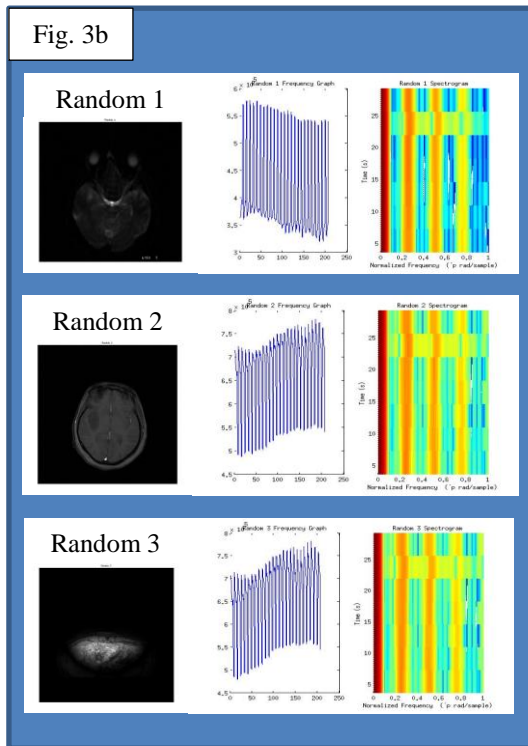
The algorithm applied to samples 1, 2, and 3 were tested against 15 images. 3 of them were from its own sample set, 6 were from the other two sample sets, 3 were from the control set and 3 were random images. The results were fairly the same from sample to sample.

- These are the results from sample 1. Shown are the image tested, its frequency graph, as well as its spectrogram.



- In sample 2, we can see that images don't match up exactly with the reference, however there are enough weights under 300,000 that we can consider sample 2 akin to 1
- In g-l, sample 3 and the control had all their weights under 300,000. Each of their spectrograms correlates to the one below.





7) Discussion

- This data has only been used through a reference set of sample 1. The data from the reference sets of samples 2 and 3 were run and found to be redundant to report as they come to the same overall conclusion as sample 1's reference.
- As we can see in figure 3a compared to 3b, test MR images of breasts correlate differently with the reference set of breasts, than do the test MR images of regions other than the breast.
- We can see in the frequency graphs of the randoms, that none of the weight values met the filter requirement of 300,000. This is indicated by the fact that the line in each of their graphs never touch zero in the way that the samples and control did.
- We also see in the spectrograms, that high correlation tends to have more red expressed in the image. This of course excludes sample 3 and the control due to none of the weights they had were over 300,000. For the random MR images, they do not have such an expression of red.

- The control did not live up to expectation by having a correlation no different than the samples. Since this control's purpose was to detect whether the algorithm was detecting lumps or something else, I can conclude that this algorithm is not sophisticated enough to detect breast lumps.

8) Conclusion

Based on the results we can make the following claims:

- The algorithm as it exists now cannot distinguish between a breast MR image afflicted with a lump and one that is not.
- To achieve the initial goal, I believe that more samples, and more controls need to be collected with doctor approval that the images they select do or do not have lumps in them.
- What the algorithm can do is automatically detect whether or not the MR image is of a breast if the reference is of a breast. I can assume that with similar techniques, we could make the reference set of a brain and when presented with MR images, it would select brain images to have a high correlation.

9) Closing Remarks

- From this project I have learned much about what is required to process medical images. I learned how to work with those images in their innate format dicom. What I take away from this semester is a stronger foundation to continue on my path in biomedical imaging.

10) Acknowledgements

There are a handful of people responsible for leading me to complete this project:

- I. First, to Michael Sheinfeld. Much of what I did here in regard to learning the out of class tools needed for this algorithm was done thanks to this man.
- II. Second, to Dr. Balzano for her abundance in patience and teaching ability to convey what was necessary for me to implement the in class tools learned for this algorithm.
- III. Finally, to the Graduate Student Instructors. With their direction, I was able to tell them what I wanted to do for this project and they gave me the idea to look into the concept of Eigenfaces.

I thank you all.

11) References

- Michael Sheinfeld, (2014, March 17th) *Matlab Central File Exchange*. [Online] <http://www.mathworks.com/matlabcentral/fileexchange/45915-eigenfaces-algorithm>
- A. Oppenheim *et al.* *Discrete-Time Signal Processing*, 3rd ed. ISBN-13:978-0-13-198842-2