Topic: PCA for Object Character Recognition

Project Report for CSE 535 Numerical Computation

Fall 2016

School of Computer Science and Engineering

California State University San Bernardino

Team Members:

Nashawna Olton

Beverly Abadines

Elham Alzaki

**Date: December 8th, 2016**

**I. Introduction**

Principal component analysis (PCA) uses linear transforms that is based on statistical techniques that are used. Using this method in image processing delivers a powerful tool for data analysis and pattern recognition. It is often used in signal and image processing as a method for data compression or data dimension reduction. There are numerous algorithms based on multivariate analysis or neural networks that can perform PCA on a given data set.

The purpose of this project is to apply PCA for Object Character Recognition (OCR) in order to convert images of characters into texts of characters. This can allow the meaning and values of words to be transferred from an image into a file, providing for the ability to store, edit, and process real world images. This project also iteratively developed larger data to test the effects of data compression towards larger dimensions.

**II. Related Work**

The advantages of using Prochaka’s article offers valuable information on principal component analysis (PCA) in image processing, complete with examples of image compression and reduction. This article takes a scientific approach to the advantages of using eigenvalues and eigenvectors. In Prochaka’s article it deals with two distinct applications of PCA which is image color reduction and object detection. In the process, Prochaka includes writing that is verifiable and scientifically consistent with current works. The disadvantages of using this article is that it mainly discusses the theoretical part of image processing and not implementation.

The advantages of using Matlab’s article offers valuable information on principal component analysis (PCA) that uses the method of transforming a number of correlated variables into a smaller number of uncorrelated variables. It contains examples of single image decomposition. This article takes a scientific approach similar to fourier analysis of orthogonal based vectors. In Matlab’s article it deals with applications of unsupervised training for images. The disadvantages of using this article is that it offers source code that has multiple errors.

The advantages of using Lee’s article offers information on principal component analysis (PCA) and singular value decomposition (SVD). It uses the variance that is retained and maximization of variance and the minimization of least square. It contains graphs that explains eigenvectors and pca examples. This article takes a mathematical approach to solving pca with eigen value decomposition. The disadvantages of using this article is that it does not offer detailed explanation of the source code.

**III. Methodology Framework**

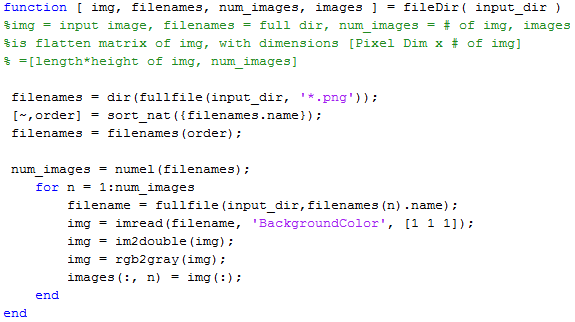
To test the capabilities of PCA for OCR this study used three data sets with varying dimensions, but identical procedures using PCA. All the steps are summarized into the development of a 2 dimensional dataset, centralization of data, discovery of eigenvalues and eigenvectors, and test images and outputted results.

It is worth noting that only the first data set was able to display supplemental figures and graphs due to its small size.

**Developing the Data**

The initial sample contained 15 images of 3 unique letters, ‘A’, ‘B’, and ‘C’ written in five different fonts, which were Calibri, AgencyFb, BellMT, MS Gothic, and Arial Narrow. The purpose of the varying fonts was to promote *some* variability towards each letter, while promoting similarities from the letter’s innate shapes, respectively. To avoid long computational times, the images were kept small with dimension sizes of 38 x 24, (height x width).

The data was then compressed from 3 dimensions of 38 x 24 x 15 (Height x Width x Amount of Images), into a matrix called “images,” which had 2 dimensions of 38\*24 x 15 (Pixel Dimension x Amount of Images) and was flattened into grayscale. This was executed by the fileDir.m.



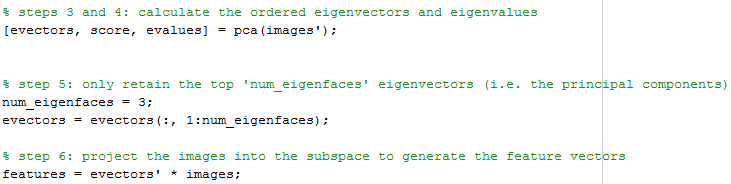
**Centralizing the Data**

The mean from all data dimensions was subtracted from the dataset, so that its new mean was zero.

https://lh6.googleusercontent.com/N6W9frCAmojiVnKIUq_5vK2XaxkgWbSyrpYm35DGoSbpqtumnfrcezeEewPhjJXYEwG2aQF_z_IG__JUaUVxUSPUWriQhjqIFrEIKTyypGH8Lxma1xVtpqSqGBWg_rpgxyFGJCDd

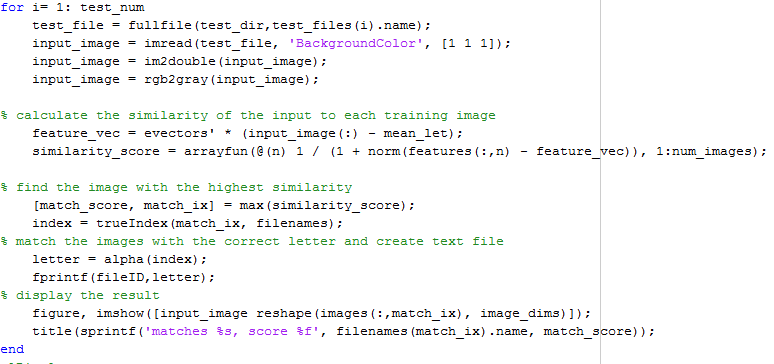
**Finding Eigenvalues and Eigenvectors**

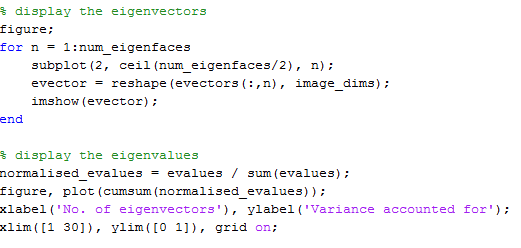
Matlab’s pca(x), which implemented Single Value Decomposition (SVD), centered the data and calculated the principal component coefficients (evectors), score, and variances (evalues), where row scores represented image observations and column scores represented image components. The largest evalues was then set as the new basis and the corresponding evector was the principal component of the data. The num\_eigenfaces was editable and retained the maximum variance of the data, emphasizing the similarities and differences of the normalized, image features.

****

**Testing Images and Outputting Results**

Testable images were compared with the transformed data, feature\_vec, to discern which images from the data were most similar to them. Afterwards the selected, corresponding image from the data was then mapped to a character based off of its index from the folder using trueIndex(X) and alpha(X). The letters were then outputted into a text file called small\_text\_message.txt and displayed figures with the tested images next to the selected data image. Additionally, because the sample was small, it was appropriate to display the eigenvectors and a graph to display the relationship between eigenvalues and variance.

****

****

**Bigger Data**

After testing a small dataset, a larger dataset was implemented to include all 26 letters of the English alphabet as well as period to serve as a delimiter for spaces. This new sample was a square matrix of 30 x 30 with 135 images written in either Calibri, AgencyFb, BellMT, Garamond, and Times New Roman. Lastly, a dataset with larger pixel dimensions of 45 x 45 with 135 images were implemented with the same five types of fonts. For both of these data, the outputted result was sent to large\_text\_message.txt and did not display the figures or graphs from the smaller dataset due to long computational times.

**IV. Results and Discussions**

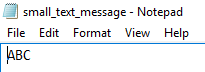
This project applied PCA’s ability to identify patterns in data through image compression in order to recognize and convert images of letters into texts. The initial goal was to create simple datasets of images to serve as training images for the algorithm and serve as references towards testable images. Once the algorithm shared some successful results, more complicated and larger datasets were implemented to fully recognize PCA’s capabilities for OCR. Altogether, there were three data sets with varying dimensions and amount of images.

**Data Results for Different Sets**

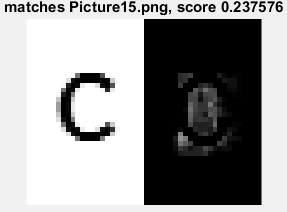
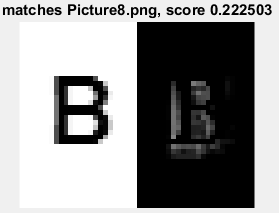
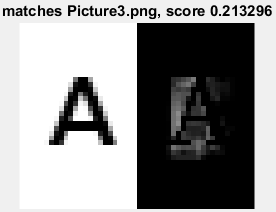
Dataset ONE (38\*24 x 15)

The data with (38\*24 x 15) contained 15 images of 3 unique letters, ‘A’, ‘B’, and ‘C’ written in five different fonts, which were Calibri, AgencyFb, BellMT, MS Gothic, and Arial Narrow as reiterated from the Methods. This set had three test images in the order of ‘A,’ ‘B,’ and ‘C’. From the outputted text file and displayed figures, PCA calculated 3 out of 3 letters correctly.  The eigenvectors and graph provided supplemental information on variance.

**Small\_text\_message.txt**

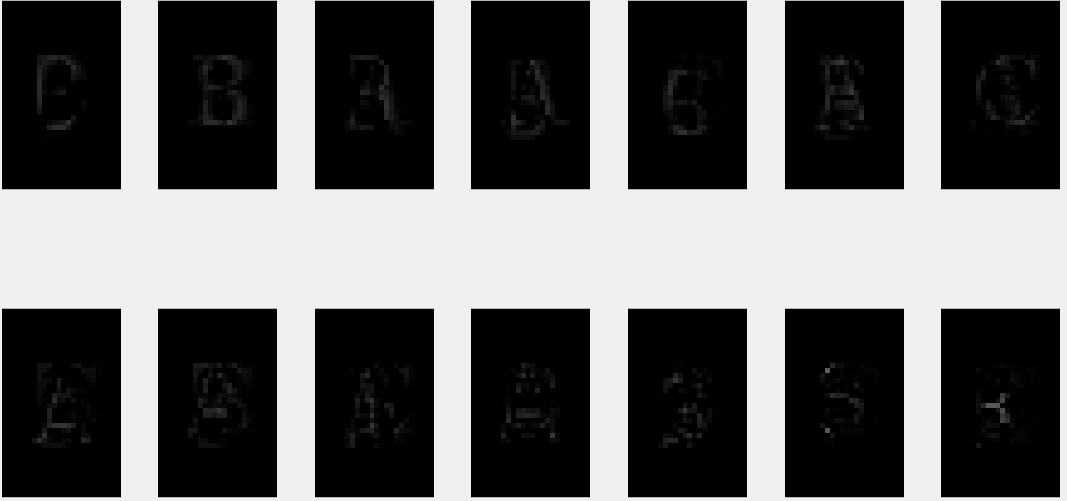


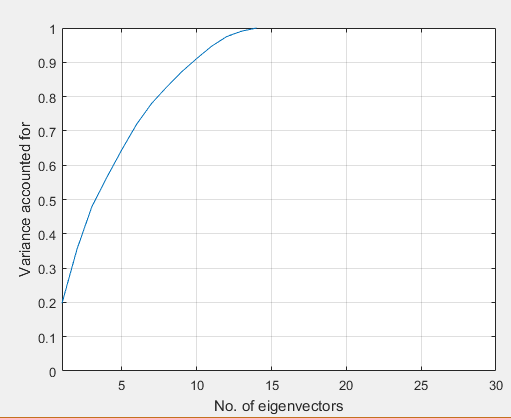
**Figures of Tested Images Next to Selected Images**



**Supplemental Graphs**

**Eigenvectors**





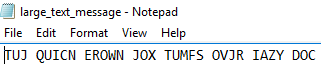
**Variance vs. Number of Eigenvectors**

This further emphasizes PCA’s characteristic which demonstrates how the highest variability, which is associated to its eigenvalues, constitutes to the strongest pattern of the data.

Dataset TWO (30\*30 x 135)

This composed of all 26 letters of the English alphabet with a period as a delimiter for spaces written in five different fonts as mentioned in the Methods. This had a pixel dimension of 900 and could only be tested using the outputted text file. There were 39 test images ordered to form the phrase, “THE QUICK BROWN FOX JUMPS OVER LAZY DOG” to test all letters in the alphabet. The outputted results were not as accurate as the smaller dataset and was able to calculate only 29 out of the 39 letters and spaces correctly.

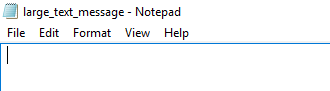
**Large\_text\_message.txt for 30 x 30**

****

Dataset THREE (45\*45 x 135)

This dataset was tested and was composed similarly to the second dataset, but contains 2025 pixels. The output of this dataset was significantly less accurate than the smaller data, categorizing all the images as spaces.

**Large\_text\_message.txt for 45 x 45**

****

**Analysis**

The three types of data had varying results, despite using the exact same algorithm. Dataset three is arguably inconclusive and may require debugging and interpretation of the data type representation. The first two datasets were able to discern over 74% of the letters correctly, which demonstrates the reliability and accuracy of PCA.

Our weakness was that we did not initially use a square matrix, which could not have eigenvectors and eigenvalues, but luckily Matlab’s PCA innately uses SVD, which can be used for non-square matrices, relying on the fact that it can form matrices that have non-zero values along the diagonal. Additionally, compared to previous a lab with Non-negative matrix factorization (NMF), NMF was able to process data by 1024 x 69 without losing information.

The project’s strengths were the incorporation of raw data sets produced by the authors, the application of PCA characteristics, and the testing of various dimension sets. Because we created our own test images, we were able to create and alter image dimensions to adjust towards matrix operations. Our study recognized the relationship of many of the PCA characteristics, such as eigenvalues, eigenvectors, variance, orthogonality, and data compression. Unlike other studies, which only projected PCA’ data representation, our study calculated the most similar image from the data given an inputted image *and* tested the effects of data compression towards higher dimensions. Although PCA provides lower dimensionality representation of more complicated multivariable information, it does experience loss of information, which can have minor or major effects depending on the type and size of the data.

**V. Problems and Improvements**

Problems

1. Image dimensions were not consistent. Each image consisted of different dimensions and it did not allow the use of pca
2. Images were initially not grayscale
3. Background color was causing interference within images and caused our image eigenvalues to be zero
4. Image type were not all the same. Some images were of type double and some were of type integer
5. Calculating the ordered eigenvectors and eigenvalues
6. Test image was not being recognized when running the algorithm
7. Files were unordered when accessed because they were char data types, not numeric
8. Index of selected, most matching images did not correspond to the correct file

Solved Problem

1. Created each image using microsoft publisher to match dimensions of each image
2. Implemented rgb2gray() in the code to fix grayscale image
3. Researched and implemented background color function into imread()
4. Implemented im2double() to make each image of type double
5. Changing princomp() function to the pca() function since it was outdated
6. Changing the directory’s location of test image
7. Implemented sort\_nat()
8. Wrote trueIndex() and alpha() for corresponding files and characters

**VI. Benefits**

Knowledge

1. Learning the statistical procedure of principal component analysis and singular value decomposition.
2. Research using covariance matrix by studying the formula and learning how to implement the code using matlab.
3. Learning dimensionality reduction by mapping data to uncorrelated dataset.
4. Expanding our knowledge in linear transformations.
5. Learned how to train images by reshaping and matching.

Skills and Techniques

1. Coding in matlab
2. Troubleshooting
3. Expanding research abilities

**VII. References**

Lee, S. (2016, December 4). *PCA and SVD*. Lecture. Retrieved December 4, 2016.

M. (2011). Eigenfaces face recognition (MATLAB). Retrieved December 04, 2016, from https://blog.cordiner.net/2010/12/02/eigenfaces-face-recognition-matlab/

PRINCIPAL COMPONENT ANALYSIS IN IMAGE PROCESSING - USTC. (n.d.). Retrieved December 4, 2016, from http://www.bing.com/cr?IG=5AEA644A469146018F8818D7DAA5A452&CID=37283D33D4DF6CC11F0B34D2D5EE6DD6&rd=1&h=illtbGxXv-jqQu9GMAavx3gPctBVDQOMW5jGac7f2Iw&v=1&r=http://staff.ustc.edu.cn/~zwp/teach/MVA/pcaimg.pdf&p=DevEx,5038.1

Smith, L. I. (2002, February 26). A tutorial on Principal Components Analysis. 1-27.