Virginia Tech

Handwritten Digit Classification and Reconstruction of Marred Images Using Singular Value Decomposition

Author:
Andy Lassiter

Supervisor:
Dr. Serkan Gugercin

May 8, 2013

Abstract

Singular Value Decomposition (SVD) is considered to be the holy grail of matrix factorizations. Here an SVD method is used to classify handwritten digits and to aid in the recovery of marred facial images from an ensemble of similar images.

1 Introduction

Matrix decomposition is a transformation of a given matrix into a product of matrices such as LU, QR, Cholesky, and Singular Value Decomposition. Of the prominent matrix factorizations Singular Value Decomposition (SVD) is considered to have the greatest generality and wide range of applications including signal processing, data mining, matrix approximation, least squares fitting of data, principal component analysis, pattern recognition, determining the rank, range, and null space of a matrix, etc. This paper explores the pattern recognition properties of SVD by classifying handwritten digits and using eigenfaces to recover marred facial images. An overview of SVD is first presented followed by the classification of handwritten digits and the reconstruction of marred facial images.

2 Singular Value Decomposition

2.1 SVD

The **singular value decomposition** (**SVD**) of any $m \times n$ matrix A is

$$A = U\Sigma V^T, \tag{1}$$

where U and V are $m \times n$ and $n \times n$ matrices, respectively, with $U^TU = V^TV = I$ and Σ is an $n \times n$ diagonal matrix. The diagonal entries of Σ are the singular values of A and are arranged such that

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0.$$

The columns of U and V are called the *left-singular vectors* and *right-singular vectors* of A, respectively, for the corresponding singular value. The two norm of a matrix is also determined from the singular values with

$$\|A\|_2 = \sigma_1.$$

The singular vectors U and V give the four fundamental subspaces of a matrix. Let $\mathcal{R}(A)$ and $\mathcal{N}(A)$ denote the range and null space of A. As summarized by Eldén in [9, p. 62], the subspaces revealed by SVD are:

1. The singular vectors $u_1, u_2, \ldots, u_r$ are an orthonormal basis for $\mathcal{R}(A)$ and
$$rank(A) = dim(\mathcal{R}(A)) = r.$$
2. The singular vectors $v_{r+1}, v_{r+1}, \ldots, v_n$ are an orthonormal basis for $\mathcal{N}(A)$ and
$$dim(\mathcal{N}(A)) = n - r.$$
3. The singular vectors $v_1, v_2, \ldots, v_r$ are an orthonormal basis for $\mathcal{R}(A^T)$.
4. The singular vectors $u_{r+1}, u_{r+1}, \ldots, u_m$ are an orthonormal basis for $\mathcal{N}(A^T)$

with r being the rank of the matrix. Thus the rank of A is also the number of non-zero singular values.

2.2 Low-rank Optimal Matrix Approximation in the 2-norm using SVD

There are situations where we need to approximate a matrix with one of a lower rank such as storing large matrices and compressing images. To achieve this approximation SVD is used. Matrix A written in summation form is

$$A = \sum_{k=1}^{n} \sigma_k u_k {v_k}^T.$$

Examining the matrix in this form shows that the zero singular values add no extra information to the matrix. The summation can then be reduced to

$$A = \sum_{k=1}^{r} \sigma_k u_k {v_k}^T$$

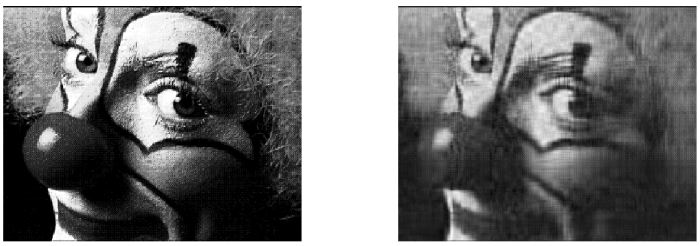
where r is the rank of A, i.e. the number of nonzero singular values. A fundamental question is to find a rank-α matrix $\tilde{A}$ with $\alpha < r$ so that $\|A - \tilde{A}\|_2$ is minimized, i.e. what is the optimal low-rank approximation to A in the 2-norm. The answer is automatically given by the SVD [9, p. 63]:

$$\tilde{A} = \sum_{k=1}^{\alpha} \sigma_k u_k {v_k}^T$$

where $\alpha < r$. Then

$$\|A - \tilde{A}\|_2 = \sigma_{\alpha+1}.$$

To illustrate the importance of this approximation an image of a clown, Figure 1, is approximated using SVD. The original image was of rank 200 while the compressed image was of rank 20. We have reduced the rank by a factor of 10, and the image of the clown is still clearly visible, though not a sharp as the original. The original image was 200×320, and 64000 entries needed to be stored. For the reduced rank image $200 \times 20 + 20 \times 20 + 20 \times 320 = 10800$ entries need to be stored, a reduction of 83%.



(a) Rank 200 (b) Rank 20

Figure 1: Image compression of a clown.

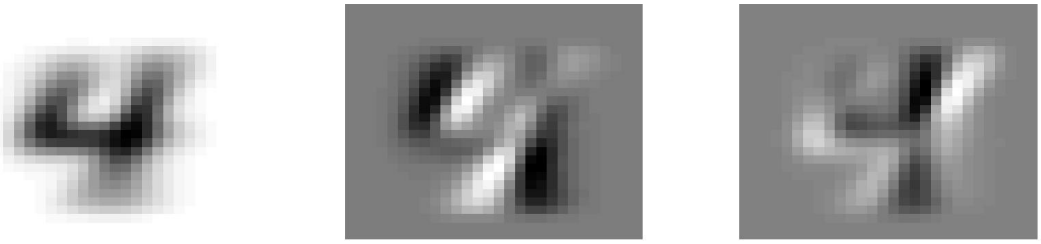
3 Classification of Handwritten Digits

Handwriting recognition has many practical applications ranging from signature verification, postal-address interpretation, and bank-check processing. In the survey of handwriting recognition by R. Plamondon and S. N. Srihari [15] handwriting recognition is defined as "the task of transforming a language represented in its spatial form of graphical marks into its symbolic representation." This recognition process can be categorized into two approaches: on-line and off-line. In an on-line approach the image is stored as a function of time showing the strokes taken by the writer.In an off-line approach only the final image is available. Classification using SVD is an off-line approach. To study the SVD approach, handwritten digits are classified following the technique presented by Eldén [9, p. 115]. The process and results are summarized below.

3.1 Theory and Algorithm

Each image of a handwritten digit can be considered as an $m \times m$ matrix where each entry in the matrix is a grayscale pixel value. The columns of each image are stacked to form a column vector of size $m^2 \times 1$. All the stacked images of a single digit are concatenated to form a matrix $A_j \in \mathbb{R}^{m^2 \times n}$, with n being the number of training images for a particular digit and $j = 0, 1, \ldots, 9$ being the particular digit. Finding the SVD of A_j will give the fundamental subspaces of a digit. The left singular vectors u_i form an orthonormal basis in the "image space" of a digit, and are referred to as a "singular image" (Figure 2). Classification based on SVD relies on the assumption that an unknown digit can be better approximated in one particular basis of singular images than in the bases of other digits. This is done by computing the residual between the unknown digit and a linear combination of singular images:

$$\min_{\alpha_i} \left\| z - \sum_{i=1}^{k} \alpha_i u_i \right\| = \min_{\alpha} \|z - U_k \alpha\|_2 ,$$



(a) First Singular Image (b) Second Singular Image (c) Third Singular Image

Figure 2: Singular images of a four from the MNIST database

where z is an unknown digit, u_i the singular images of a particular digit,k the number of bases used to approximate the image, and $U_k = (u_1 \quad u_2 \quad \dots \quad u_k)$. Ideally this minimum is equal to zero, and thus the solution to the problem is give by $\alpha = U_k^T z$. The residual is then given by:

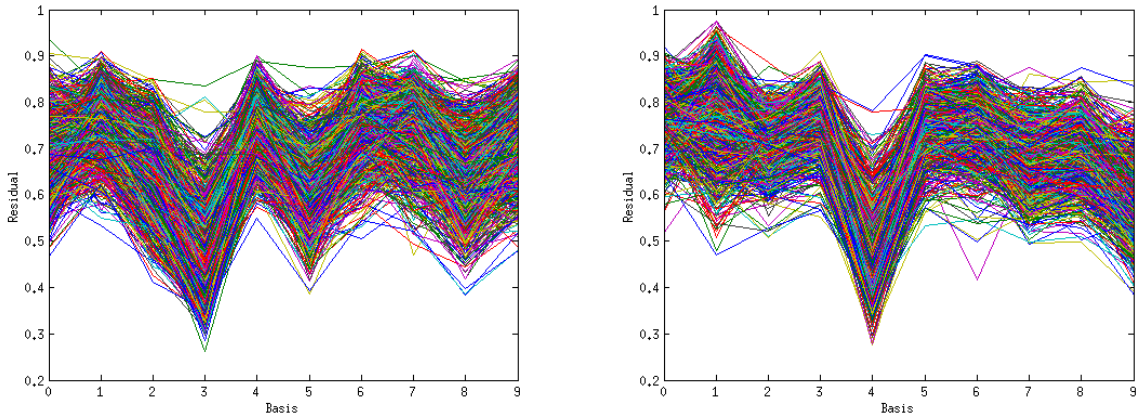
$$\left\|(I - U_k {U_k}^T)z\right\|_2 . \tag{2}$$

Handwritten digit classification can thus be separated into two steps: training and classification. First the SVD of each training set of known digits is computed, yielding ten sets of left-singular vectors. An unknown digit is then classified against ten residuals using equation (2) with the digit being classified as a member of the basis which yields the smallest residual.

3.2 Results

Two image databases were used to test the SVD classification method: the MNIST Database of Handwritten Digits and USPS Handwritten Digit Database. The MNIST database contains a training set $60,000$ and a test set of $10,000$ images of size 28×28. The database in its original form can be obtained from Yann LeCun's webpage [13]. The USPS database contains 7291 training and 2007 test images of size 16×16. The MNIST and USPS databases in MATLAB format used for this experiment were obtained from [2] [5] [6].

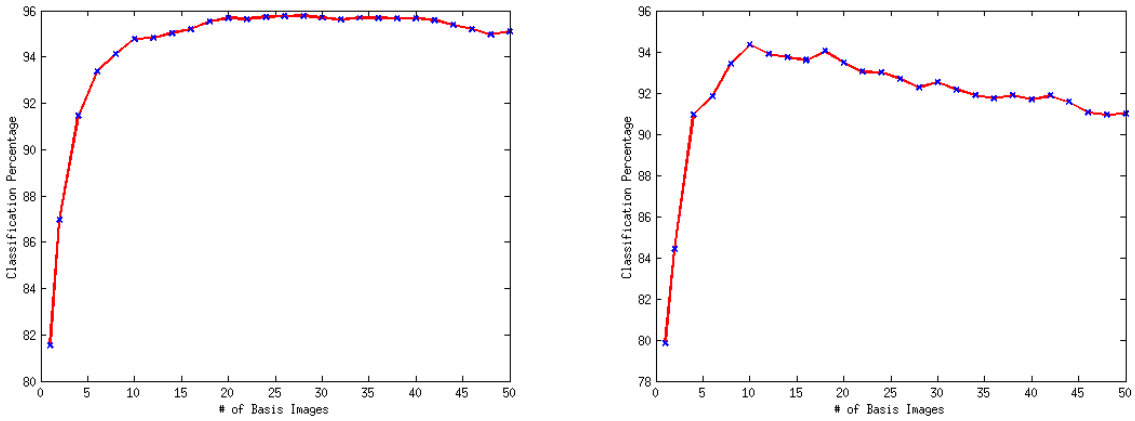
To determine whether the relative residual (Eq. 2) was sufficient for classifying unknown digits the relative residuals of the test images of the 3's and 4's were calculated using ten singular images (Figure 3). The plots show a sharp drop in the relative residual at 3 and at 4. This indicates equation (2) is a reasonable method for classification. Figure 3 also shows the similarities of 3 and 4 to other digits. The digit 3 is similar to 5's and 8's and dissimilar



(a) Test images for the digit 3.

(b) Test images for the digit 4.

Figure 3: Relative residuals of the test set of 3's and 4's from the MNIST database. Ten basis vectors were used.



(a) MNIST (b) USPS

Figure 4: Correct classification percentage as a function of the number of basis images.

1's and 4's. On the other hand 4's are similar to 9's while being dissimilar to other digits.

What needs to be determined next is the number of singular images needed to correctly classify a digit. Figure 4 shows the correct classification percentage as a function of the number of basis images. The percent correct approaches an equilibrium value of about 96% after 15 basis images for the MNIST database. At 10 basis images 95% of the digits are classified correctly. Theses five extra basis images add little to the outcome and increase the computational time so these extra bases might be unnecessary depending on the level of accuracy needed. For the USPS database a maximum value of 94% percent is reached after 10 basis images and decreases to 92% with more bases. For both databases, 10 basis images contain the need information for classification, and anything beyond that is of little value.

To further test the SVD classification algorithm, handwritten digit samples were collected from students of a math class. The students were asked to write their zip codes into a predefined grid on an iPad. The digits, shown in the appendix (Figure 9), were then transferred to a computer, normalized, and run through the classification algorithm using the MNIST singular images. The algorithm correctly classified 71% of the digits which is much lower than expected. Writing on the iPad with a finger and keeping the numbers inside the grid seemed difficult for students. For example Figure 5 shows a students zip code which strayed outside of the grid. Even though the first digit is a three, when the image was input into the computer the bottom portion of the three was cut off to make it look like a two. Subsequently the image was classified as a two. Improvements in classification would be expected if the grid was made much larger and a stylus made available for writing.

(a) A student's zip code

(b) First digit of zip code as cropped by the computer

Figure 5: Example of a student zip code which is outside of the grid. The digit, when input into the computer, looks like and was classified as a two.

4 Reconstruction of Marred Faces

There are many applications in which data is collected as an ensemble of similar data but some of this data contains gaps. For example remote sensing satellite images contain clouds, and Netflix has a list of movies you've watched and would like to suggest movies to you [1]. The clouds and movie suggestions are gaps which we would like to fill. To demonstrate this idea an image of a face marred with noise is recovered from an ensemble of similar facial images.

4.1 Eigenfaces

The reconstruction process begins in a similar manner to the classification of handwritten digits. Following the notation of [16], let $\varphi(\bar{x})$ be a scalar function representing an image of a face, with $\bar{x} = (x_1, x_2)$ a pixel location and φ the gray level at that location. An ensemble of facial images is then formed,$\{\varphi^{(n)}\}$ with $n = 1, \ldots, M$ and M the number of images. Next the average value at each pixel location is determined to find the average image.

$$\langle\varphi\rangle = \frac{1}{M}\sum_{n=1}^{M}\varphi^{(n)}.$$

A new ensemble is then formed consisting of the deviation from the mean:

$$\phi^{(n)} = \varphi^{(n)} - \langle\varphi\rangle.$$

Each image of the ensemble is transformed into a stacked column vector and concatenated as done previously with the images of handwritten digits to form the matrix ϕ. The SVD of ϕ is then determined:

$$\phi = U\Sigma V^T.$$

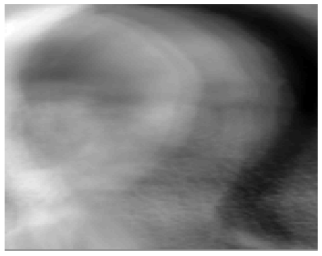
The left-singular vectors, U, will form an orthogonal basis of all facial images if given a large enough training set. Thus any face, f, whether or not it is included in the ensemble, can be approximated as

$$f \approx \sum_{n=1}^{N} a_n u_n$$

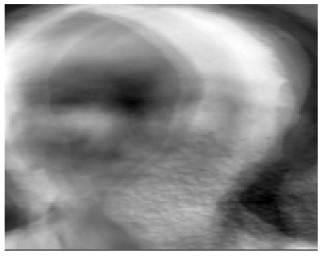
with $N < M$ and the coefficients $a_n = \langle f, u_n \rangle = f^T u_n$. The singular images in this orthogonal basis are referred to as *eigenfaces*. This method of representing faces was first reported by L. Sirovich and M. Kirby [16], and a good review of the topic is given by N. Muller, K. Magala, and B. M. Herbst [14]. To study eigenfaces the Yale Face Database was used [11]. The Yale Face Database consists of 165 grayscale images of 15 individuals. There are 11 images of each subject consisting of a different facial expression or lighting configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. The faces in this set are not normalized so a normalized set was obtained from [3] [8] [4] [7] [12]. In the original images the faces were off-center and the heads tilted slightly. In the normalized set the noses were at the center, the heads were not tilted, the images were cropped to remove empty space. A normalized set of images is very important in order for the eigenface representation to work correctly. This is illustrated in Figure 6. For the set which was not normalized the singular images look more like a ghost or an alien while the

(a) First eigenface from normalized set.

(b) Second eigenface from normalized set.



(c) First eigenface from unnormalized set.



(d) Second eigenface from unnormalized set.

Figure 6: Eigenfaces from normalized and unnormalized set.

singular images of the normalized set captured the human face rather well.

4.2 Reconstruction of Marred Faces

A marred face is our representation of a marred data set. The goal is to recover a marred facial image with the clean image not being in the ensemble but being similar to the faces in it. This is done by following the work of R. Everson and L. Sirovich in [10]. We start with our marred image:

$$\tilde{f}(\mathbf{x}) = m(\mathbf{x})f(\mathbf{x})$$

where $f(\mathbf{x})$ is the facial image we want to recover, $\tilde{f}(\mathbf{x})$ the masked face, and $m(\mathbf{x})$ is the mask with $m = 0$ being masked and $m = 1$ unmasked. Our goal is to write $\tilde{f}(\mathbf{x})$ as

$$\tilde{f}(\mathbf{x}) \approx m(\mathbf{x}) \sum_{n=1}^{N} \tilde{a}_n u_n$$

and determine the best set of coefficients $\tilde{a}_n$. These coefficients are found by minimizing the error

$$\left[\tilde{f}(\mathbf{x}) - m(\mathbf{x}) \sum_{n=1}^{N} \tilde{a}_n u_n \right]$$

which leads to

$$\langle \tilde{f}(\mathbf{x}) - m(\mathbf{x}) \sum_{n=1}^{N} \tilde{a}_n u_n, u_k \rangle_{s[\tilde{f}]}$$

where $k = 1, \ldots, N$, and the inner product is over the support of $\tilde{f}$. From this we can form two matrices:

$$M_{kn} = \langle u_k, u_n \rangle_{s[\tilde{f}]}$$

and

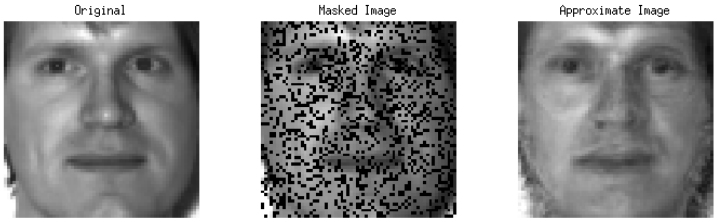
$$y_k = \langle \tilde{f}, u_k \rangle_{s[\tilde{f}]}.$$

The coefficients $\tilde{a}_n$ are thus obtained by solving

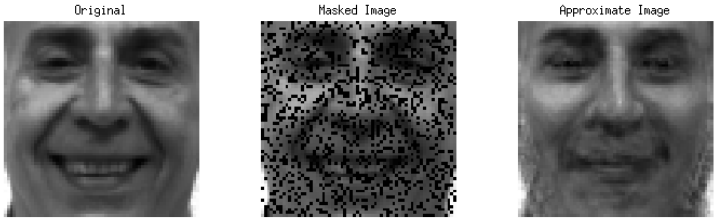
$$M = \tilde{a}y,$$

and we obtain our unmasked image

$$f(\mathbf{x}) = \sum_{n=1}^{N} \tilde{a}_n u_n.$$



(a)

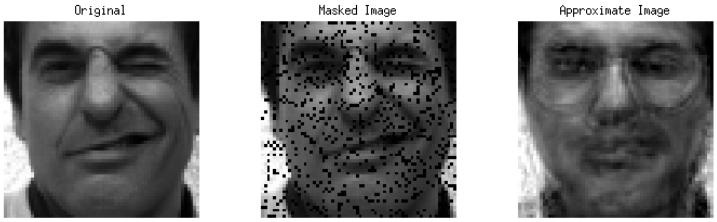


(b)

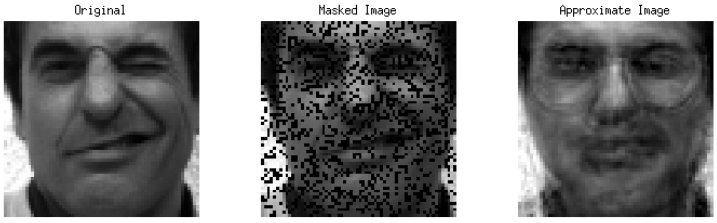
Figure 7: Example of recovering a marred faces with 20% masks.

4.3 Results

Faces from the normalized Yale Face Database were masked and the images recovered. Figure 7 shows two faces marred with a 20% mask and the recovered image. The first face recovered (Figure 7(a)) looks similar to the original. The face is neutral, he is not smiling or frowning, not winking or wearing glasses, and the light is at center. This is well represented by the ensemble as the first two eigenfaces in Figure 6 show. The second face (Figure 7(b)) looks somewhat similar to the original but there was difficulty recovering the man's smile. There are no teeth visible in the recovered image. Teeth must not be a well represented feature of the ensemble. Figure 8 shows another example of a face with a 10% and 20% mask. The recovered images look quite different from the original. The mouth has been blurred and glasses added to the subject. The winking, which also alters the man's mouth, must not be well represented by the ensemble. These examples show that to recover a marred image, or any marred data, the image must be well represented by the ensemble. More facial images are needed to better characterize and recover a face.



(a) 10% mask.



(b) 20% mask.

Figure 8: Example of recovering a marred face.

5 Summary

The singular images of handwritten digits and eigenfaces demonstrate the power of singular value decomposition for pattern recognition. Very few singular images were need to capture the fundamental features of a large data set. They enabled us to recover masked features of an image, and as the ensemble of data of data becomes larger, the quality of the recovered image will improve. Though SVD was used for image processing purposes, it can easily be expanded to other areas where pattern recognition and marred data recovery is needed.

References

[1] Robert M. Bell, Yehuda Koren, and Chris Volinsky. The BellKor 2008 Solution to the Netflix Prize. `http://www.netflixprize.com/assets/ProgressPrize2008_BellKor.pdf`. Accessed: 5/5/2013.

[2] Deng Cai. Matlab Codes and Datasets for Subspace Learning (Dimensionality Reduction). `http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html`. Accessed: 2/22/2013.

[3] Deng Cai. Popular Face Data Sets in Matlab Format. `http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html`. Accessed: 2/22/2013.

[4] Deng Cai, Xiaofei He, and Jiawei Han. Spectral Regression for Efficient Regularized Subspace Learning. In *Proc. Int. Conf. Computer Vision (ICCV'07)*, 2007.

[5] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S. Huang. Graph Regularized Nonnegative Matrix Factorization for Data Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

[6] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S. Huang. Graph Regularized Nonnegative Matrix Factorization for Data Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

[7] Deng Cai, Xiaofei He, Jiawei Han, and Hong-Jiang Zhang. Orthogonal Laplacianfaces for Face Recognition. *IEEE Transactions on Image Processing*, 15(11):3608–3614, 2006.

[8] Deng Cai, Xiaofei He, Yuxiao Hu, Jiawei Han, and Thomas Huang. Learning a Spatially Smooth Subspace for Face Recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning (CVPR'07)*, 2007.

[9] Lars Eldén. *Matrix Methods in Data Mining and Pattern Recognition*. Society for Industrial and Applied Mathematics, 2007.

[10] R. Everson and L. Sirovich. Karhunen–Loève Procedure for Gappy Data. *J. Opt. Soc. Am. A*, 12(8):1657–1664, Aug 1995.

[11] A. S. Georghiades. Yale Face Database. `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`, 1997.

[12] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face Recognition Using Laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(3):328–340, 2005.

[13] Yann LeCun and Corinna Cortes. The MNIST Database of Handwritten Digits. `http://yann.lecun.com/exdb/mnist/`. Accessed: 2/22/2013.

[14] N. Muller, L. Magaia, and B. Herbst. Singular Value Decomposition, Eigenfaces, and 3D Reconstructions. *SIAM Review*, 46(3):518–545, 2004.

[15] R. Plamondon and S.N. Srihari. Online and Off-line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.

[16] L. Sirovich and M. Kirby. Low-dimensional Procedure for the Characterization of Human Faces. *J. Opt. Soc. Am. A*, 4(3):519–524, Mar 1987.

Appendices

A Figures

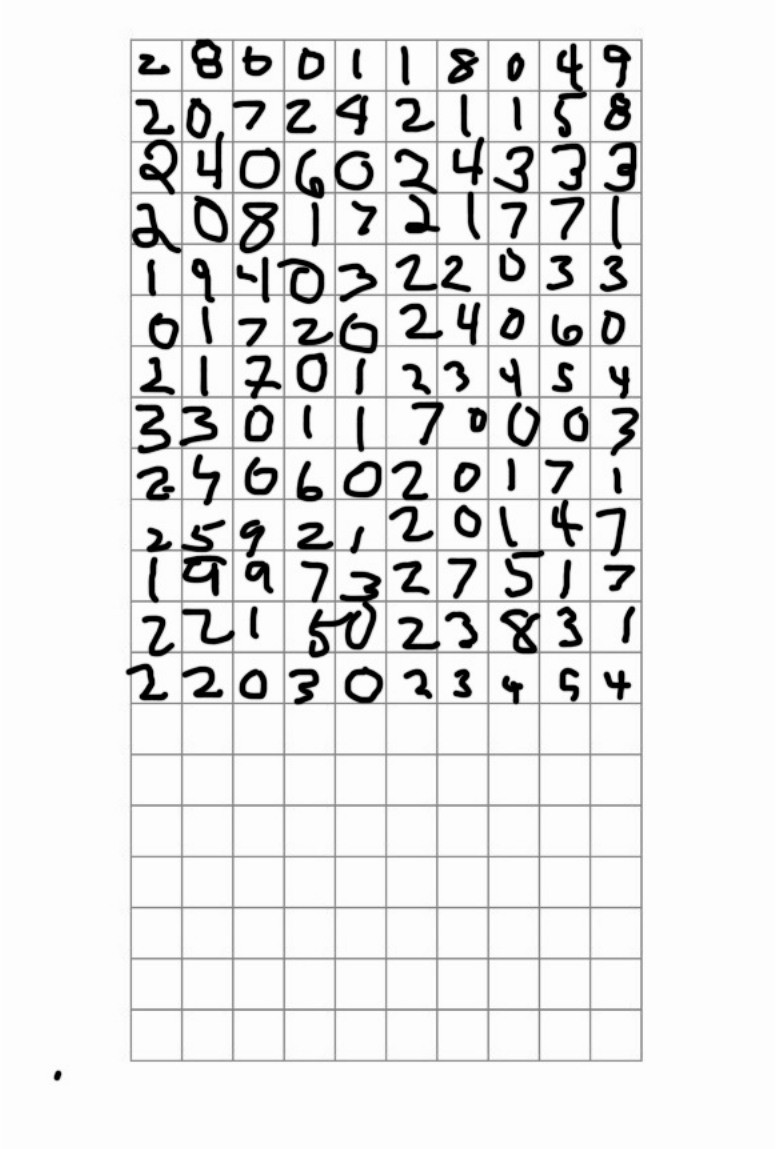


Figure 9: Zip codes collected from a Virginia Tech math class