Fast Classification of Handwritten Digits Using 2D-DCT Based Sparse PCA

Darya Ismailova and Wu-Sheng Lu
Electrical and Computer Engineering
University of Victoria
Victoria, BC, Canada
Email: ismailds@uvic.ca, wslu@uvic.ca

***Abstract*—We propose to address the handwritten digits recognition (HWDR) problem by using a two-dimensional (2-D) discrete cosine transform (DCT) based sparse principal component analysis (PCA) algorithm for fast classification. The gain of processing speed is achieved by utilizing the ability of 2-D DCT for energy compaction and signal decorrelation. The proposed algorithm was applied to the mixed national institute for standards and technology (MNIST) database of handwritten digits to demonstrate that when incorporated into the conventional PCA, the 2-D DCT helped reduce the dimension of the input data by 75%. As a result of the dimensionality reduction, the proposed algorithm is 35.7% faster for HWDR than the conventional PCA without sacrificing recognition accuracy.**

***Index Terms*—DCT; 2D-DCT; PCA; classification.**

I. Introduction

It is well known that the optical character recognition (OCR) problem is a simple vision task yet difficult to accomplish with machines [1], and in this connection, research for handwritten digit recognition (HWDR) by machine learning (ML) techniques has stayed active in the past several decades [2]-[7]. The sustained interest in HWDR is primarily due to its broad applications in bank check processing, postal mail sorting, automatic address reading, and mail routing, etc. In these applications, both accuracy and speed of digit recognition are critical indicators of system performance. In a machine learning setting, we are given a training data set consisting of a number of samples of handwritten digits, each belongs to one of the ten classes $\{\mathcal{D}_j, j = 0, 1, \ldots, 9\}$ where class $\mathcal{D}_j$ collects all data samples labeled as digit j. The HWDR problem seeks to develop an approach to utilizing these known data classes to train a multi-class classifier so as to recognize handwritten digits outside the training data. The primary challenge arising from the HWDR problem lies in the fact that handwritten digits (within the same digit class) vary widely in terms of shape, line width, and "style", even when they are normalized in size and properly centralized. In this regard, it is interesting to look at reference [7] which describes a popular database, known as MNIST, of handwritten digits, and includes a list of performance (in terms of accuracy) records achieved by ML techniques.

Classification of multi-class data can be accomplished by ML methods that are originally intended for classifying binary-class data using the so-called *one-versus-the-rest* approach [8, p.338]. Alternatively, there are ML techniques that deal with multi-class data directly. A representative method in this class is based on principal component analysis (PCA) [9]. At its origin PCA is a statistical procedure that is used as a tool in data analysis and it can be thought of as revealing the internal structure of the data in terms of the variance in the data. As the data size rapidly grows, contemporary use of PCA also covers dimensionality reduction in connection to detection and classification for large-scale data. Reference [3] presents detailed performance analysis and comparisons of several ML techniques for the HWDR problem, including PCA-based linear and polynomial classifiers.

In this paper, we present a new PCA-based classifier for HWDR with improved recognition speed. The improvement in speed is achieved by sparsifying handwritten digits as digital images using two-dimensional (2-D) discrete cosine transform (DCT), and retaining only significant DCT coefficients so as to reduce the dimensionality of the input space. The proposed sparse PCA-based classifier was applied to the MNIST database of handwritten digits where the classifier was trained using a total of 12,000 handwritten digits from its training data and was tested by applying the classifier to 10,000 handwritten digits from its testing data. The result was a 35.7% improvement in recognition speed relative to that of the conventional PCA-based classifier. To our surprise, the proposed classier also offers a slightly improved recognition accuracy of 96.26% over what was achieved by the conventional PCA-based classifier (96.21%) although the dimension of the input space was reduced by 75% over the original input space. The paper is organized as follows. In Sec. 2, we review the conventional PCA and explain how it can be utilized for multi-category classification. In Sec. 3, the 2-D DCT-based sparse PCA is proposed and its application to multi-category classification problem is outlined. In Sec. 4, the proposed algorithm is examined by applying it to the real-world data from the MNIST handwritten digit database. Concluding remarks are given in Sec. 5.

II. PCA for Multi-Category Classification

Also known as the Karhunen-Loeve expansion [10], PCA is one of the most successful techniques for feature extraction and data representation. Kirby and Sirovich [10] applied the method for the identification and classification of human

978-1-4673-7788-1/15/$31.00 ©2015 IEEE

faces. In [11], Turk and Pentland introduced the concept of "eigenfaces" based on PCA for face recognition. Application of PCA in handwritten digits classification and evaluation of its performance in terms of recognition rate, speed, and memory requirements relative to other ML techniques such as soft-margin support vector machine (SVM) and LeNet 5 etc. can be found in [3]. To describe our 2-D DCT-based sparse PCA, in this section we outline the conventional PCA and explain how a PCA-based multi-category classifier works.

Suppose we are given n real-valued data samples $\{\boldsymbol{x}_i, i = 1, 2, \ldots, n\}$ with each $\boldsymbol{x}_i$ being a column vector of dimension m. We define average vector and covariance matrix of this data set $\{\boldsymbol{x}_i, i = 1, 2, \ldots, n\}$ as

$$\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i, \quad \boldsymbol{C} = \frac{1}{n-1}\sum_{i=1}^{n} (\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T$$

respectively. Note that $\boldsymbol{C}$ is positive semidefinite, hence its singular value decomposition (SVD) is identical to its eigen-decomposition

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T$$

where $\boldsymbol{U} = [\boldsymbol{u}_1\, \boldsymbol{u}_2 \ldots \boldsymbol{u}_m]$ is an orthogonal matrix, $\boldsymbol{S} = diag\{\sigma_1, \sigma_2 \ldots \sigma_m\}$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m$, and $\boldsymbol{u}_i$ is $\boldsymbol{C}$'s ith eigenvector associated with ith eigenvalue σ_i. By retaining only the K largest eigenvalues and the associated eigenvectors, we obtain an L_2-optimal rank-K approximation of covariance matrix $\boldsymbol{C}$ as

$$\boldsymbol{C} \approx \boldsymbol{U}_K \boldsymbol{S}_K \boldsymbol{U}_K^T \tag{1}$$

with

$$\boldsymbol{U}_K = [\boldsymbol{u}_1\, \boldsymbol{u}_2 \ldots \boldsymbol{u}_K], \quad \boldsymbol{S}_K = diag\{\sigma_1, \sigma_2 \ldots \sigma_K\}$$

The usefulness of (1) may be understood from two different but related perspectives as follows.

(a) Since $\boldsymbol{C}$ is the variance for $\{\boldsymbol{x}_i - \overline{\boldsymbol{x}}, i = 1, 2, \ldots, n\}$, the first K eigenvectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_K\}$ capture the most significant variations between the individual data points. For this reason they are referred to as the first K *principal components* of the data set. It follows that each point $\boldsymbol{x}_i - \overline{\boldsymbol{x}}$ is well represented by its projection onto the K-dimensional subspace spanned by $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_K\}$, i.e.,

$$\boldsymbol{z}_i = \boldsymbol{U}_K^T (\boldsymbol{x}_i - \overline{\boldsymbol{x}}) \tag{2}$$

From (2), we see that the point $\boldsymbol{x}_i$ can be approximated by a point in that subspace, plus the average $\overline{\boldsymbol{x}}$, as

$$\boldsymbol{x}_i \approx \boldsymbol{U}_K \boldsymbol{z}_i + \overline{\boldsymbol{x}}$$

In this way, the original data set $\boldsymbol{X} = [\boldsymbol{x}_1\, \boldsymbol{x}_2 \ldots \boldsymbol{x}_n]$ in R^m is approximately equivalent to the data set $\boldsymbol{Z} = [\boldsymbol{z}_1\, \boldsymbol{z}_2 \ldots \boldsymbol{z}_n]$ in space R^K, plus a single average vector $\overline{\boldsymbol{x}}$. Note that dimension K of the subspace spanned by $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_K\}$ is typically much less than dimension m of the space of input samples, and this is how PCA can be of use for dimensionality reduction in data processing.

(b) Now let us consider the case of supervised multi-category classification with a training set that consists of L data classes of the form $\mathcal{D}_j = \{(\boldsymbol{x}_i^{(j)}, y_i), i = 1, 2, \ldots, n_j\}$ for $j = 0, 1, \ldots, L-1$, where each sample $\boldsymbol{x}_i^{(j)}$ is associated with the same label y_j. The problem here is to classify a point $\boldsymbol{x}$ outside the training data to one of the L classes. To this end, PCA is applied to each data class $\{(\boldsymbol{x}_i^{(j)}, y_i), i = 1, 2, \ldots, n_j\}$ by first computing the average vector and covariance matrix as

$$\overline{\boldsymbol{x}_j} = \frac{1}{n_j}\sum_{i=1}^{n_j} \boldsymbol{x}_i^{(j)}, \quad \boldsymbol{C}_j = \frac{1}{n_j - 1}\sum_{i=1}^{n_j} (\boldsymbol{x}_i^{(j)} - \overline{\boldsymbol{x}_j})(\boldsymbol{x}_i^{(j)} - \overline{\boldsymbol{x}_j})^T$$

then performing eigen-decomposition of $\boldsymbol{C}_j$ as

$$\boldsymbol{C}_j = \boldsymbol{U}_j \boldsymbol{S}_j \boldsymbol{U}_j^T$$

and approximating it by

$$\boldsymbol{C}_j \approx \boldsymbol{U}_K^{(j)} \boldsymbol{S}_K^{(j)} \boldsymbol{U}_K^{(j)T}$$

where $\boldsymbol{U}_K^{(j)} = [\boldsymbol{u}_1^{(j)}\, \boldsymbol{u}_2^{(j)} \ldots \boldsymbol{u}_K^{(j)}]$ consists of K eigenvectors of $\boldsymbol{C}_j$ that are associated with the K largest eigenvalues and $\boldsymbol{S}_K^{(j)}$ is a diagonal matrix that collects the K largest eigenvalues in descent order. In least-squares sense, the L data classes $\{\boldsymbol{x}_i^{(j)}, i = 1, 2, \ldots, n_j\}$ for $j = 0, 1, \ldots, L-1$ are well represented by L reduced "data" sets $\{\overline{\boldsymbol{x}_j}, \boldsymbol{U}_K^{(j)}\}$ for $j = 0, 1, \ldots, L-1$ respectively, where $\overline{\boldsymbol{x}_j}$ denotes the average character of the jth class $\{\boldsymbol{x}_i^{(j)}, i = 1, 2, \ldots, n_j\}$, and $\boldsymbol{U}_K^{(j)}$ defines a K-dimensional subspace that resides in R^m for satisfactory representation of the variations of the individual members in the jth class from its average $\overline{\boldsymbol{x}_j}$. The dimension reduction becomes substantial when $K \ll m$ which is indeed the case in many applications including the HWDR problem, as will be demonstrated in Section 4. The classification of a testing point $\boldsymbol{x}$ is carried out as follows: (i) Project point $\boldsymbol{x} - \overline{\boldsymbol{x}_j}$ into the jth data class, i.e. compute $\boldsymbol{z}_j = \boldsymbol{U}_K^{(j)T}(\boldsymbol{x} - \overline{\boldsymbol{x}_j})$ and approximate point $\boldsymbol{x}$ in the jth class as $\widehat{\boldsymbol{x}_j} = \boldsymbol{U}_K^{(j)} \boldsymbol{z}_j + \overline{\boldsymbol{x}_j}$; (ii) Compute the Euclidean distance between $\boldsymbol{x}$ and its approximation $\widehat{\boldsymbol{x}_j}$, i.e., $e_j = \|\boldsymbol{x} - \widehat{\boldsymbol{x}_j}\|$ for $j = 0, 1, \ldots, L-1$; (iii) Point $\boldsymbol{x}$ is classified to class j^* if e_{j^*} reaches the minimum among $\{e_j, j = 0, 1, \ldots, L-1\}$.

III. 2-D DCT-Based Sparse PCA

A. 2-D DCT

For image-related data, 2-D DCT [12] has proven effective in compressing the data and has indeed been adopted in standard JPEG [13] for compression, storage, and transmission of digital images. Given a digital image of size N by N represented by its light intensity $\{x(i,j), i, j = 1, 2, \ldots, N\}$, the 2-D DCT of the image is a 2-D array of the same size, denoted by $\{D(k,l), k, l = 1, 2, \ldots, N\}$ where

$$D(k,l) = \frac{2\alpha(k)\alpha(l)}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} x(i,j) \cdot \cos\left(\frac{(2i+1)k\pi}{2N}\right)\cos\left(\frac{(2j+1)l\pi}{2N}\right)$$

with

$$\alpha(k) = \begin{cases} \frac{1}{\sqrt{2}}, \text{ for } k = 0 \\ 1 \text{ for } k \neq 0 \end{cases}$$

A salient property of 2-D DCT is its ability of energy compaction, meaning that a digital image can be well represented by a relatively small number of DCT coefficients. As an example Fig.1 shows (a) a handwritten digit of size 28 by 28 from MNIST database, (b) its 2-D DCT, and (c) a plot of the 784 2-D DCT coefficients as a 1-D sequence obtained by zigzag scanning the "image" in (b) from top left to bottom right. Using Frobenius norm E_5 of the image in (a) to represent the total energy of the digit, E_5 was found to be 8.2637. Since 2-D DCT is known to preserve image's energy, the 2-norm of the entire DCT coefficients also equals to 8.2637. The 2-norms of the first 6.25% (i.e. the first 49), 12.5% (i.e. the first 98), and 25% (i.e. the first 196) 2-D DCT coefficients were found to be 6.5415, 7.2688, and 7.9307, respectively, representing 79.16%, 87.96%, and 95.97% of the image's total energy.

B. 2-D DCT-Based Sparse PCA

The main point where the 2-D DCT-based sparse PCA differs from the conventional PCA is the use of 2-D DCT to reduce the dimension m of the input data space without severe degradation in recognition accuracy. Consider an image-related training data $\{\boldsymbol{x}_i, i = 1, 2, \ldots, n\}$ such as MNIST where each $\boldsymbol{x}_i$ is a column vector of dimension m obtained by stacking the columns of an image on top of one another. To take the advantage of 2-D DCT, we reshape vector $\boldsymbol{x}_i$ to its original image size to which 2-D DCT is applied. The 2-D DCT coefficients are then converted to a 1-D sequence by zigzag scanning the coefficients (see Fig. 1c) and retaining the first r DCT coefficients to construct a column vector $\boldsymbol{d}_i$ of length r. The truncation (hence dimension reduction) is justified by the energy compaction property of 2-D DCT and the dimension reduction from m to r can be significant if $r \ll m$. For the HWDR problem, for example, r can be as small as $m/4$ for fast recognition with excellent accuracy. We remark that the use of a small portion of the DCT coefficients simply means the use of a *sparse* 2-D DCT where all components, except the first r of them, are set to zero to approximately represent the original image.

The treatment of training data $\{\boldsymbol{x}_i, i = 1, 2, \ldots, n\}$ using 2-D DCT leads to a reduced data set $\{\boldsymbol{d}_i, i = 1, 2, \ldots, n\}$. The key equation for PCA, namely

$$\boldsymbol{C} \approx \boldsymbol{U}_K \boldsymbol{S}_K \boldsymbol{U}_K^T$$

holds where

$$\overline{\boldsymbol{d}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{d}_i, \quad \boldsymbol{C} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{d}_i - \overline{\boldsymbol{d}})(\boldsymbol{d}_i - \overline{\boldsymbol{d}})^T$$

and

$$\boldsymbol{S}_K = diag\{\sigma_1, \sigma_2, \ldots, \sigma_K\}, \quad \boldsymbol{U}_K = [\boldsymbol{u}_1\, \boldsymbol{u}_2 \ldots \boldsymbol{u}_K]$$

are constructed by the K largest eigenvalues of $\boldsymbol{C}$ and the associated eigenvectors (known as principal components), just like in the conventional PCA with the difference in the size of the $\boldsymbol{u}_i$'s: it is now reduced from m to r. Consequently, computing the projection $\boldsymbol{z}_i = \boldsymbol{U}_K^T(\boldsymbol{d} - \overline{\boldsymbol{d}})$ and the approximating point $\boldsymbol{d}_i \approx \boldsymbol{U}_K \boldsymbol{z}_i + \overline{\boldsymbol{d}}$ can be done more efficiently relative to the conventional PCA.

It is rather straightforward to extend the proposed 2-D DCT-based sparse PCA to the supervised multi-category classification problem discussed in Sec. 2, where the training set consists of L data classes of the form $\mathcal{D}_j = \{(\boldsymbol{x}_i^{(j)}, y_j), i = 1, 2, \ldots, n_j\}$ for $j = 0, 1, \ldots, L-1$. As a result, we obtain a modified PCA-based algorithm for multi-category classification which may be summarized step by step as follows.

Algorithm 1 (2-D DCT-based sparse PCA for multi-category classification)

Input: Training data $\mathcal{D}_j = \{(\boldsymbol{x}_i^{(j)}, y_j), i = 1, 2, \ldots, n_j\}$ for $j = 0, 1, \ldots, L-1$; target dimension of reduced input space r; number K of principal components to be retained; and testing data $\mathcal{T}$.

Step 1: Apply 2-D DCT to $D_j, j = 0, 1, \ldots, L-1$ to obtain reduced data set $R_i = \{(\boldsymbol{d}_i^{(j)}, y_j), i = 1, 2, \ldots, n_j\}$ of dimension r.

Step 2: For $j = 0, 1, \ldots, L-1$ compute

$$\overline{\boldsymbol{d}_j} = \frac{1}{n}\sum_{i=1}^{n_j} \boldsymbol{d}_i^{(j)}, \boldsymbol{C}_j = \frac{1}{n_j - 1}\sum_{i=1}^{n_j}(\boldsymbol{d}_i^{(j)} - \overline{\boldsymbol{d}_j})(\boldsymbol{d}_i^{(j)} - \overline{\boldsymbol{d}_j})^T$$

and the K eigenvectors $\boldsymbol{U}_K^{(j)} = [\boldsymbol{u}_1^{(j)}\, \boldsymbol{u}_2^{(j)} \ldots \boldsymbol{u}_K^{(j)}]$ that are associated with the K largest eigenvalues of $\boldsymbol{C}_j$.

Step 3: For each vector $\boldsymbol{x}$ from test data $\mathcal{T}$:

(i) Compute vector $\boldsymbol{d} \in R^r$ by applying 2-D DCT to the image constructed from $\boldsymbol{x}$ and retaining the r most significant DCT coefficients (refer to Sec. 3.A);

(ii) Perform projections for $j = 0, 1, \ldots, L-1$;

(iii) Compute approximating points $\widehat{\boldsymbol{d}_j} = \boldsymbol{U}_K^{(j)} \boldsymbol{z}_j + \overline{\boldsymbol{d}_j}$ for $j = 0, 1, \ldots, L-1$;

(iv) Compute $e_j = \|\boldsymbol{d} - \widehat{\boldsymbol{d}_j}\|$ for $j = 0, 1, \ldots, L-1$;

(v) Classify point $\boldsymbol{x}$ to class j^* if e_{j^*} reaches the minimum among $\{e_j, j = 0, 1, \ldots, L-1\}$.

IV. Application to Handwritten Digit Recognition

A. The MNIST Database

The MNIST handwritten digit database [7] has been popular for performance evaluation of ML algorithms and methods as applied to the HWDR problem. It contains training data of 60,000 handwritten digits, each is a gray-scale image of size 28 by 28, and a separate testing data of 10,000 handwritten digits, each image is of the same size. Associated with each digit is a label as the true meaning which the digit represents. Table I shows the cardinality distribution of the ten sets of digits representing the ten Arabic numerals. Each data sample from the database is a vector of length 784 obtained by stacking the columns of the aforementioned image. The one-to-one correspondence between the vector and matrix representations of the digit makes it straightforward to visualize an individual digit from the database. Fig.2 displays sample handwritten digits from the training and testing data.

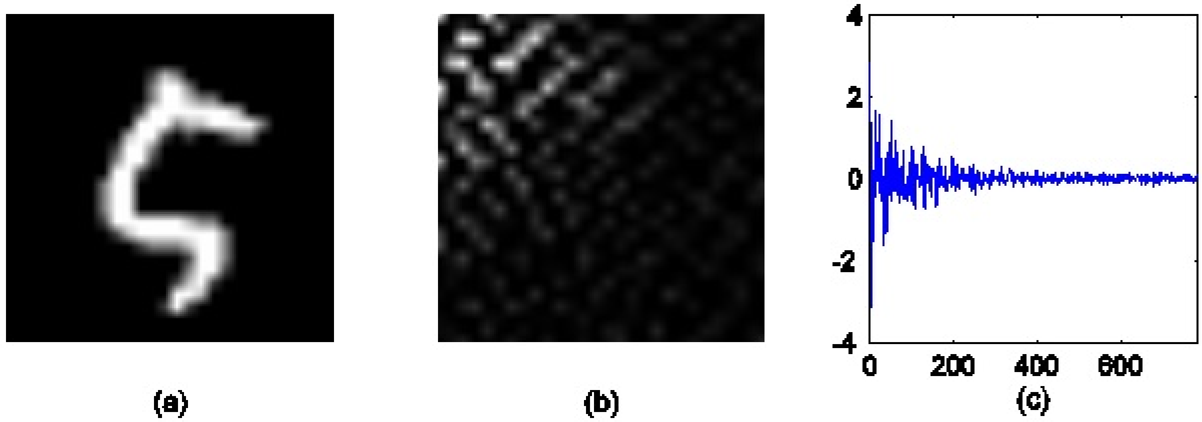
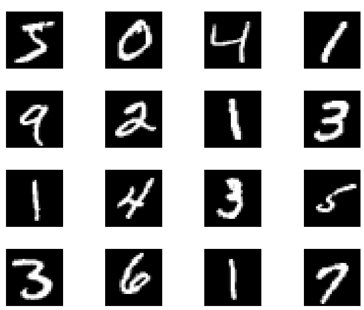


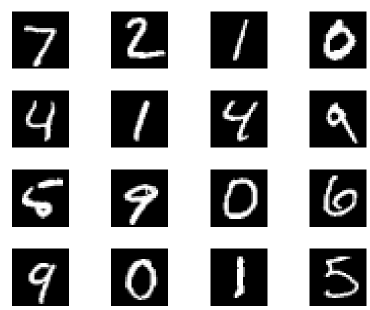
Fig. 1: (a) An example digit from MNIST database, (b) the 2-D DCT of the image of size 28 by 28 in (a), and (iii) the 784 DCT coefficients arranged by zigzag scanning the "image" in (b) from top left towards bottom right.

TABLE I: Distribution of MNIST sets of digits

Digit	0	1	2	3	4	5	6	7	8	9	Total
Training set	5923	6742	5958	6131	5842	5421	5918	6265	5851	5949	60000
Test set	980	1135	1032	1010	982	892	958	1028	974	1009	10000



(a) Training set



(b) Test set

Fig. 2: Typical images from the MNIST database

B. Generation of Input Data for HWDR

The input data is composed of ten sets of training data $\mathcal{D}_j = \{(\boldsymbol{x}_i^{(j)}, y_j), i = 1, 2, \ldots, n_j\}$ for $j = 0, 1, \ldots, 9$, where each $\mathcal{D}_j$ contains a total of n_j digits representing the same numeral j. In our experiments, n_j was set to 1200 for all sets and, for a fixed j, $\{\boldsymbol{x}_i^{(j)}, i = 1, 2, \ldots, 1200\}$ were selected at random from those in the training data that collects all the digits representing numeral j. It is important to stress that, like any ML algorithms based on statistical learning theory, the performance of the proposed algorithm varies slightly when the aforementioned random selection of the data is initiated from a different random seed. For this reason we refer the procedure of implementing an ML algorithm utilizing the input data that is generated with a specific random seed as one *run*. Consequently, by performing multiple runs, each with a distinct random seed, results representing the average performance or best performance can be obtained.

C. Performance Evaluation of 2-D DCT Based Sparse PCA

Preliminary experiments were conducted to identify appropriate ranges for parameters r and K (see Sec. 3 for the definitions of these parameters). Satisfactory recognition results were obtained when the above parameters fall within in the ranges $180 \leq r \leq 400$ and $22 \leq K \leq 31$, respectively. Choosing a value of parameter r is critical as r is closely related to how fast HWDR is performed. A smaller r yields a faster classifier, but using an r too small degrades recognition accuracy. In our simulations, r was set to 196, representing a 75% reduction of the dimension of the input data which is 28 x 28 = 784. To the authors surprise, the first 196 DCT coefficients were shown to be able to capture sufficient characteristics of the digits as the proposed sparse PCA found no loss in recognition accuracy relative to the conventional PCA classifier (see below for details).

With n_j = 1200 and r = 196, both the proposed sparse PCA and conventional PCA based classifiers were applied to the data from the MNIST database. For each classifier, we conducted a total of 200 runs: in each run a distinct random seed was set for selecting 1200 sample digits at random from the database for each of the ten numerals, then with K = 22, 23, . . . , 31, ten classifiers were constructed and applied to recognize the 10,000 testing digits from the database. The best conventional PCA classifier was obtained when the random seed was set to 48 and a total of K = 25 principal components were utilized, achieving a recognition rate of 96.21%, while the best sparse PCA was obtained when the random seed was set to 154 and a total of K = 26 principal components were utilized, yielding a recognition rate of 96.26%. While the two classifiers offer practically the same recognition accuracy, on a 64-bit Windows-based desktop PC with an i7-3770 CPU@3.4GHz and 32G RAM, in average it took the sparse PCA classifier 0.93 ms to recognize one digit, versus 1.446 ms per digit as required by the conventional PCA classifier, representing a 35.7% improvement in recognition speed.

We conclude this section with a remark that better recognition accuracy for handwritten digits can be achieved by using more advanced ML techniques such as several variants of support vector machines and LeNet 5 [7], however the improvement of recognition accuracy is achieved at the cost of much increased complexity. In this regard the proposed sparse PCA classifier provides an attractive alternative for real-time HWDR.

V. Conclusion

A 2-D DCT-based sparse PCA classifier for handwritten digit recognition has been proposed. The ability of 2-D DCT to compress image-related signals allows a significant dimensionality reduction of the input space, while capturing characteristic information of handwritten digits. By applying it to the MNIST database, the sparse PCA classifier is shown to perform HWDR considerably faster than the conventional PCA classifier without sacrificing recognition accuracy.

References

[1] S. Haykin, *Neural Networks - A Comprehensive Foundation*, Macmillan College Publishing Co., 1994.

[2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, pp. 541–551, 1989.

[3] Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, and V. Vapnik, "Comparison of learning algorithms for handwritten digit recognition", *Int. Conf. on Artificial Neural Networks*, pp. 53–60, 1995.

[4] G. E. Hinton, P. Dayan, and M. Revow, "Modeling the manifolds of images of handwritten digits," *IEEE Trans. Neural Networks*, vol. 8, no. 1, pp. 65–74, Jan. 1997.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, pp. 2278–2324, Nov. 1998.

[6] F. Lauer, C. Y. Suen, and G. Bloch, "A trainable feature extractor for handwritten digit recongnition," *Pattern Recognition*, vol. 40, pp. 1816–1824, 2007.

[7] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits," available online: *http://yann.lecun.com/exdb/mnist*

[8] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[9] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., Springer, 2002.

[10] M. Kirby and L. Sirovich, "Application of Karhunen-Loeve procedure for the characterization of human faces," *IEEE Trans. PAMI*, vol. 12, pp. 103–108, 1990.

[11] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," *Proc. CVPR*, pp. 586-591, 1991.

[12] K. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications,* Academic Press, 1992.

[13] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, 3rd ed., Springer, 1993.