Handwritten Digits Recognition Using PCA of Histogram of Oriented Gradient

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

***Abstract*—This paper presents a multiclass classifier based on principal component analysis (PCA) of histogram of oriented gradient (HOG) for accurate and fast recognition of handwritten digits. HOG is known as an effective feature descriptor for computer vision and image processing, and PCA has shown its ability for fast multiclass recorgenition. By combining PCA with HOG, the PCA-of-HOG based classifier is developed. The proposed algorithm was applied to the MNIST database of handwritten digits to demonstrate its performance in comparison with classifiers based on PCA of raw input data.**

I. Introduction

In the handwritten digits recognition (HWDR) problem, we are given a training data set consisting of a number of samples of handwritten digits, each belongs to one of the ten classes $\{D_j, j = 0,\ 1,\ \ldots,\ 9\}$ where class D_j collects data samples that are labeled as digit j. We seek an approach to utilizing the available labeled data to train a multi-class classifier to recognize handwritten digits *outside* the training data. Research for HWDR using machine learning (ML) techniques has stayed active in the past several decades [1]–[7] due primarily to its broad applications in postal mail sorting and routing, automatic address reading, and bank check processing, etc. The challenge arising from the HWDR problem lies in the fact that handwritten digits (within a given digit class) vary widely in shape, line width, and "style", even when they are normalized in size and properly centralized. Reference [6] describes a popular database, known as MNIST, of handwritten digits, and includes a list of performance (in terms of accuracy) records achieved by various ML techniques. Recognition 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]. 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]. PCA is a statistical procedure that can be thought of as revealing the internal structure of the data in terms of the variance in the data. Contemporary use of PCA also covers dimensionality reduction in connection to detection and classification for large-scale data. Reference [2] presents 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 accuracy. The improvement is achieved by incorporating a pre-processing module into the system to extract effective features from the input data. It is well known that applying pre-processing transformations is often advantageous compared to using raw input data directly in ML algorithms [10]. To be specific, *histogram of oriented gradient* (HOG) [11] of the training data is employed to replace the data set itself in the PCA. Based on the PCA of HOG, an algorithm is developed for HWDR to achieve performance improvement in accuracy. We demonstrate the effectiveness of the HOG-PCA classifier by applying it to recognize handwritten digits from MNIST database.

The paper is organized as follows. In Sec. II, HOG is illustrated in detail using MNIST digits as examples. In Sec. III, we present a brief review of PCA and explain how it can be utilized for multi-category classification. An algorithm based on PCA of HOG for multi-category classification problems is then outlined. In Sec. IV, the proposed algorithm is examined by applying it to HWDR problem for data from MNIST database. Concluding remarks are given in Sec. V.

II. Histogram of Oriented Gradients

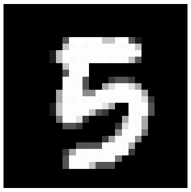
Proposed in [11], the histogram of oriented gradients (HOG) is a feature descriptor for object detection in computer vision and image processing. It has since been popular for its simplicity, desirable invariance properties and performance stability. In this section, the construction principle of HOG is illustrated.

For clarity of presentation, HOG is illustrated using gray-scale digital images of size 28 by 28 so that the analysis is consistent throughout the paper and is directly applicable to the experimental studies for HWDR reported in Sec. IV.

Let $D(x, y)$ be such an image, which represents normalized light intensity of the image at pixel (x, y). An example of image D from MNIST database is displayed in Fig. 1(a). To define the HOG of the image, we divide D into a total of 196 non-overlapping sub-images of size 2 by 2, which are called $cells$ of the image. Thus there are 14 rows and 14 columns of cells. We denote the cell in the ith row and jth column by $C(i, j)$.

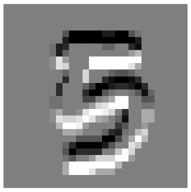
Next, we form image blocks, each as a 2-cell by 2-cell sub-image, so a block is a 4 by 4 sub-matrix. The important difference between the cells and blocks is that the cells are non-overlapping, while the blocks overlap and here is how they are constructed. Starting from the 4 cells in the upper-left corner of the image, the blocks are formed row by row, with the last block constructed using the 4 cells at the bottom-right corner. The first block, denoted by $B(1, 1)$, is a square of size 4 by 4 which consists of the 4 cells from the upper-left corner of the image, namely the cells $\{C(1, 1),\ C(1, 2),\ C(2, 1),\ C(2, 2)\}$. The next block, denoted by $B(1, 2)$, is also a square of size 4 by 4, which has 50% overlap with block $B(1, 1)$ and consists

978-1-5386-0700-8/17/$31.00 ©2017 IEEE

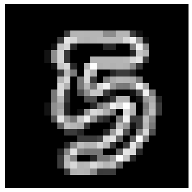


(a) A 28 x 28 image D(x,y)

(b) Derivative of D(x,y) w.r.t.x



(c) Derivative of D(x,y) w.r.t.y



(d) Magnitude of gradient of D(x,y)

Fig. 1. (a) A 28 by 28 image $D(x,y)$, (b) $\nabla_x D(x,y)$ as an image, (c) $\nabla_y D(x,y)$ as an image, and (d) $\|\nabla D(x,y)\|$ as an image.

of the 4 cells $\{C(1,2),\ C(1,3),\ C(2,2),\ C(2,3)\}$. The rest of the image blocks involving the first and second rows of cells can be similarly defined, with the last block being $B(1,13)$ which has 50% overlap with block $B(1,12)$ and consists of the 4 cells $\{C(1,13),\ C(1,14),\ C(2,13),\ C(2,14)\}$. The second row of blocks involves the second and third rows of cells and it starts from block $B(2,1)$ which has 50% overlap with block $B(1,1)$ and consists of the 4 cells $\{C(2,1),\ C(2,2),\ C(3,1),\ C(3,2)\}$. The remaining blocks $B(2,2),\ldots,B(2,13)$ are defined in a similar manner. This formation process continues until block $B(13,13)$ consisting of cells $\{C(13,13),\ C(13,14),\ C(14,13),\ C(14,14)\}$ is formed. Evidently, there are 13 rows and 13 columns of overlapping blocks, thus a total of 169 blocks. The manner in which the image blocks are formed immediately implies that each inner cell (i.e. a cell that does not contain boundary pixels) is involved in four blocks, and each boundary cell (except the cells at the images corners) is involved in two blocks. This fact that a given cell has its presence in multiple image blocks is important because it explains why the HOG (to be defined next) is insensitive to slight image deformation of position and angle.

The gradient $\nabla D(x,y)$ of image $D(x,y)$ is a vector field over the image's domain and is defined by

$$\nabla D(x,y) = \begin{bmatrix} \partial D/\partial x \\ \partial D/\partial x \end{bmatrix}$$

For digital images, however, the partial derivatives must be approximated by partial differences along the x and y directions, respectively, as

$$\frac{\partial D(x,y)}{\partial x} \approx \frac{D(x+\delta,y) - D(x-\delta,y)}{2\delta} \tag{1a}$$

$$\frac{\partial D(x,y)}{\partial y} \approx \frac{D(x,y+\delta) - D(x,y-\delta)}{2\delta} \tag{1b}$$

In practice, the approximated gradient in (1) over the entire image domain is evaluated by two-dimensional discrete convolution as

$$\frac{\partial D(x,y)}{\partial x} \quad \approx \quad D(x,y) \otimes h_x \tag{2a}$$

$$\frac{\partial D(x,y)}{\partial y} \quad \approx \quad D(x,y) \otimes h_y \tag{2b}$$

where the kernels h_x and h_y that implement the partial difference operations are given by

$$h_x = [-1\ 0\ 1],\ h_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \tag{2c}$$

In what follows, the discrete gradient of image $D(x,y)$ is denoted by $\{\nabla_x D, \nabla_y D\}$. As an example, the components of the discrete gradient, $\nabla_x D$ and $\nabla_y D$, for the image in Fig. 1(a) are illustrated in Fig. 1(b) and (c), respectively. The gradient provides two important pieces of information about the images local structure, namely its magnitude $\|\nabla D(x,y)\|$ and angle $\Theta(x,y)$ which are computed as

$$\|\nabla D(x,y)\| = \sqrt{(\nabla_x D)^2 + (\nabla_y D)^2} \tag{3a}$$

$$\Theta(x,y) = \tan^{-1}\left(\frac{\nabla_y D}{\nabla_x D}\right) \tag{3b}$$

Fig. 1(d) shows the magnitude of $\nabla D(x,y)$ for the image in Fig. 1(a). The arctangent in (3b) takes the sign of the gradient into account, i.e., it is the four-quadrant arctangent in the range between $-\pi$ and π. We stress that adopting signed gradient (as opposed to using unsigned gradient) is largely application dependent and the favour of signed gradient here has to do with the nature of the HWDR problem in that distinguishing $\tan^{-1}(a/b)$ from $\tan^{-1}((-a)/(-b))$ turns out to be beneficial.

To define the HOG of an image, we begin by dividing the angle range $(-\pi,\pi)$ evenly into b bins with the first bin occupying range $-\pi$ to $-\pi+\Delta$ where $\Delta = 2\pi/b$, and the second bin having range $-\pi+\Delta$ to $-\pi+2\Delta$, and so on. The 169 image blocks defined earlier are arranged as a sequence $\{B(1,1),\ \ldots,\ B(1,13),\ B(2,1),\ \ldots,\ B(2,13),\ \ldots,\ B(13,1),\ldots,\ B(13,13)\}$. Now we associate each block $B(i,j)$ with a column vector $\boldsymbol{h}_{i,j}$ of length b, which is initially set to $\boldsymbol{h}_{i,j} = \mathbf{0}$, and we associate each component of $\boldsymbol{h}_{i,j}$ with a bin defined above. Since $B(i,j)$ is of size 4 by 4, it has 16 pairs $\{\Theta(x,y), \|\nabla D(x,y)\|\}$. Scanning the 16 pairs one by one and, for each pair $\{\Theta(x,y), \|\nabla D(x,y)\|\}$, depending on which bin the angle $\Theta(x,y)$ falls into, the magnitude $\|\nabla D(x,y)\|$ is added to the $\boldsymbol{h}$ component associated to that bin. The vector $\boldsymbol{h}_{i,j}$ so obtained is called the HOG of block $B(i,j)$ and the HOG of image D is defined by stacking all $\{\boldsymbol{h}_{i,j}\}$ in an orderly fashion as

$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{h}_{1,1} \\ \vdots \\ \boldsymbol{h}_{1,13} \\ \boldsymbol{h}_{2,1} \\ \vdots \\ \boldsymbol{h}_{13,13} \end{bmatrix} \tag{4}$$

For an image of size 28 by 28, the length of $\boldsymbol{h}$ in (4) is equal to $169 \times b$. Typically the value of b (i.e. the number of bins per histogram) is in the range 7 to 9. With $b = 7$, for example, the vector $\boldsymbol{h}$ in (4) is of length 1183. In a supervised learning problem, for a class of training data contains N images where each image $\boldsymbol{x}_i$ is associated with an HOG vector $\boldsymbol{h}_i$, the HOG feature of the data set is a matrix of size 1183 by N in the form

$$\boldsymbol{H} = [\boldsymbol{h}_1 \ \ \boldsymbol{h}_2 \ \ \cdots \ \ \boldsymbol{h}_N] \tag{5}$$

The above steps of constructing HOG of a digital image can readily be extended to images of arbitrary sizes. The difference between any two instances, if it ever exists, is merely in terms of cell/block sizes, bin number, and adoption of signed or unsigned gradient. As a feature descriptor, HOG has proven effective and found numerous applications in computer vision and image processing. On the other hand, on comparing this complexity of feature matrix $\boldsymbol{H}$ in (5) with that of the raw data set $\{\boldsymbol{x}_1, \ \boldsymbol{x}_2, \ \ldots, \ \boldsymbol{x}_N\}$, the HOGs complexity can easily exceed that of the input images.

III. PCA OF HOG

The principal component analysis (PCA) has been a successful technique for dimension reduction, feature extraction, and data representation [9], [12], [13]. Application of PCA in handwritten digits classification and performance evaluation in terms of recognition rate, speed, and memory requirement relative to other ML techniques can be found in [2]. In this section, we briefly outline the principle of PCA, then explain how a PCA-based multi-category classifier works, and present an algorithm based on the PCA of HOG for HWDR.

A. PCA

Suppose we are given n real-valued data samples $\{x_i, i = 1, \ 2, \ \ldots, \ n\}$, each $\boldsymbol{x}_i$ is a column vector of length m. We define the *mean* and *covariance* of the data set as

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

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

$$\boldsymbol{C} = \boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^T$$

where $\boldsymbol{T} = [\boldsymbol{t}_1 \ \boldsymbol{t}_2 \ \cdots \ \boldsymbol{t}_m]$ is an orthogonal matrix, $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \ \lambda_2, \ \ldots, \ \lambda_m\}$ with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$, and $\boldsymbol{t}_i$ is the eigenvector associated with λ_i. By retaining only the K largest eigenvalues and the associated eigenvectors, an L_2-optimal rank-K approximation of covariance matrix $\boldsymbol{C}$ is obtained as

$$\boldsymbol{C} \approx \boldsymbol{T}_K\boldsymbol{\Lambda}_K\boldsymbol{T}_K^T \tag{6}$$

where

$$\boldsymbol{T}_K = [\boldsymbol{t}_1 \ \boldsymbol{t}_2 \ \cdots \ \boldsymbol{t}_k], \ \boldsymbol{\Lambda}_K = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_K\}$$

Since $\boldsymbol{C}$ is the variance for $\{\boldsymbol{x}_i - \bar{\boldsymbol{x}}, \ i = 1, \ 2, \ \ldots, \ n\}$, the first K eigenvectors $[\boldsymbol{t}_1 \ \boldsymbol{t}_2 \ \cdots \ \boldsymbol{t}_K]$ capture the most significant variations between the individual data points, hence they are called the *principal components* of the data set. As such, each point $\boldsymbol{x}_i - \bar{\boldsymbol{x}}$ is expected to be well represented by its projection onto the K-dimensional subspace spanned by $[\boldsymbol{t}_1 \ \boldsymbol{t}_2 \ \cdots \ \boldsymbol{t}_K]$, i.e.,

$$\boldsymbol{z}_i = \boldsymbol{T}_K^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}}) \tag{7}$$

From (7), point $\boldsymbol{x}_i$ can be approximated by a point in that subspace plus the average $\bar{\boldsymbol{x}}$, as

$$\boldsymbol{x}_i = \boldsymbol{T}_K\boldsymbol{z}_i + \bar{\boldsymbol{x}}$$

thus the original data set $\boldsymbol{X} = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ \cdots \ \boldsymbol{x}_n]$ in R^m is approximately equivalent to data set $\boldsymbol{Z} = [\boldsymbol{z}_1 \ \boldsymbol{z}_2 \ \cdots \ \boldsymbol{z}_n]$ in space R^K, plus a single average vector $\bar{\boldsymbol{x}}$. Note that the dimension K of the subspace is typically much smaller than the dimension m of the input data, and this is how PCA achieves dimensionality reduction in data processing.

B. A PCA-Based Multi-Category Classifier

Now consider the problem of supervised multi-category classification with a training set that consists of a total of L classes of labeled data $\mathcal{D}_j = \{(\boldsymbol{x}_i^{(j)}, y_j,) \ i = 1, \ 2, \ \ldots, \ n_j\}$ for $j = 0, \ 1, \ \ldots, \ L-1$, where, within a given class $\mathcal{D}_j$, every $\boldsymbol{x}_i^{(j)}$ is associated with the same label y_j. To classify a point $\boldsymbol{x}$ outside the training data to one of the L classes, PCA is applied to each data class by first computing the mean and covariance

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

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

$$\boldsymbol{C}_j = \boldsymbol{T}_j\boldsymbol{\Lambda}_j\boldsymbol{T}_j^T$$

and approximating it by

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

where $\boldsymbol{T}_K^{(j)} = [\boldsymbol{t}_1^{(j)} \ \boldsymbol{t}_2^{(j)} \ \cdots \ \boldsymbol{t}_K^{(j)}]$ consists of K eigenvectors of $\boldsymbol{C}_j$ that are associated the K largest eigenvalues and $\boldsymbol{\Lambda}_K^{(j)}$ 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 $\{\bar{\boldsymbol{x}}_j, \boldsymbol{T}_K^{(j)}\}$ for $j = 0, \ 1, \ \ldots, \ L-1$, respectively, where $\bar{\boldsymbol{x}}_j$ denotes the mean of $\mathcal{D}_j$, and $\boldsymbol{T}_K^{(j)}$ defines a K-dimensional subspace that resides in R^m representing the variations of the individual members in class $\mathcal{D}_j$ from $\bar{\boldsymbol{x}}_j$.

The classification of a testing point $\boldsymbol{x}$ is performed as follows: (i) Project point $\boldsymbol{x} - \bar{\boldsymbol{x}}_j$ into the jth data set by computing $\boldsymbol{z}_j = \boldsymbol{T}_K^{(j)T}(\boldsymbol{x} - \bar{\boldsymbol{x}}_j)$; (ii) Approximate point $\boldsymbol{x}$ in the jth data set as $\hat{\boldsymbol{x}}_j = \boldsymbol{T}_K^{(j)}\boldsymbol{z}_j + \bar{\boldsymbol{x}}_j$; (iii) Compute the Euclidean distance $E_j = \|\boldsymbol{x} - \hat{\boldsymbol{x}}_j\|_z$ for $j = 0,\ 1,\ \ldots,\ L-1$, and (iv) Classify $\boldsymbol{x}$ to class j^* if E_{j*} reaches the minimum among $\{E_j,\ j = 0,\ 1,\ \ldots,\ L-1\}$.

C. Multi-Category Classification Using PCA of HOG

Improved multi-category classification can be accomplished by applying PCA to HOG features instead of raw training data. Again, suppose we are given L classes of m-dimensional labeled examples $\mathcal{D}_j = \{(\boldsymbol{x}_i^{(j)}, y_j),\ i = 1,\ 2,\ \ldots,\ n_j\}$ for $j = 0,\ 1,\ \ldots,\ L-1$. For each data class $\mathcal{D}_j$, we compute its HOG feature matrix $\boldsymbol{H}^{(j)}$ to which PCA is applied to build a multi-category classifier. The algorithmic steps implementing this procedure are outlined as Algorithm 1 below.

Algorithm 1 for multi-category classification using PCA of HOG

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

Step 1: Transform the raw input data $\mathcal{D}_j$ to HOG feature matrix $\boldsymbol{H}^{(j)} = [\boldsymbol{h}_1^{(j)}\ \boldsymbol{h}_2^{(j)}\ \cdots\ \boldsymbol{h}_{nj}^{(j)}]$ for $j = 0,\ 1,\ \ldots,\ L-1$.

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

$$\bar{\boldsymbol{h}}_j = \frac{1}{n_j}\sum_{i=1}^{n_j}\boldsymbol{h}_i^{(j)},$$

$$\boldsymbol{C}_j = \frac{1}{n_j - 1}\sum_{i=1}^{n_j}(\boldsymbol{h}_i^{(j)} - \bar{\boldsymbol{h}}_j)(\boldsymbol{h}_i^{(j)} - \bar{\boldsymbol{h}})^T$$

and K dominating eigenvectors $\boldsymbol{T}_K^{(j)} = [\boldsymbol{t}_1^{(j)}\ \boldsymbol{t}_2^{(j)}\ \cdots\ \boldsymbol{t}_K^{(j)}]$.

Step 3: To classify the testing vector $\boldsymbol{x}$, do

(i) Compute HOG features $\boldsymbol{h}$ of $\boldsymbol{x}$.

(ii) For $j = 0,\ 1,\ \ldots,\ L-1$, compute projection $z_j = \boldsymbol{T}_K^{(j)T}(\boldsymbol{h} - \bar{\boldsymbol{h}}_j)$; approximation $\hat{\boldsymbol{h}}_j = \boldsymbol{T}_K^{(j)}z_j + \bar{\boldsymbol{h}}_j$; Euclidian distance $E_j = \|\boldsymbol{h} - \hat{\boldsymbol{h}}_j\|_2$

(iii) Classify point $\boldsymbol{x}$ to class j^* if E_{j^*} achieves the minimum of $\{E_j \text{ for } j = 0,\ 1,\ \ldots,\ L-1\}$.

IV. Application to the HWDR problem

PCA of HOG was applied to the HWDR problem for handwritten digits from the MNIST database. The database 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. Fig. 2 shows a few sample images from the database. Associated with each digit is a label as the true meaning the digit represents. Each data sample from the database is represented as a column vector $\boldsymbol{x}_i$ of length 784 obtained by stacking the columns of the aforementioned image.

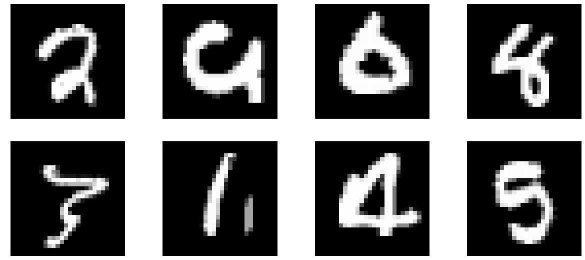


Fig. 2. Handwritten digits from MNIST.

The input data is composed of ten sets of training data $\mathcal{D}_j = \{(\boldsymbol{x}_i^{(j)}, y_j),\ i = 1,\ 2,\ \ldots,\ 1600\}$ for $j = 0,\ 1,\ \ldots,\ 9$, where each $\mathcal{D}_j$ contains a total of n_j digits representing the same label j. In our experiments, the examples in each data set $\{(\boldsymbol{x}_i^{(j)}, y_j)$, were selected at random from all the digits with label j in the training data.

Preliminary experiments were conducted to identify appropriate ranges for parameter K (see Section III for the definition of K). Satisfactory recognition results were obtained when K falls within in the range $22 \le K \le 30$. Classification experiments were carried out by applying (a) PCA to the raw training data directly as described in Sec. III.B and (b) PCA-of-HOG method outlined in Algorithm 1. For each classifier, a total of 100 runs were conducted, in each run a distinct random seed was set for selecting 1600 sample digits at random from the database for each of the ten numerals, then with $K \in \{22, \ldots, 30\}$, the two classifiers were constructed and applied to recognize the 10,000 testing digits from the database. The best performance of method (a) was a recognition rate of 96.21% when K was set to 25; while the recognition rate achieved by method (b) was 98.39% when K was set to 23 and the random seed for selecting the 16,000 training digits was set to 30 in MATLAB. It should be pointed out that the state-of-the-art performance (as far as PCA based classifiers for HWDR are concerned) was achieved at the cost of reduced speed relative to that of the PCA of raw training data. On a 64-bit Windows-based desktop PC with an i7-3770 CPU@3.4GHz and 32G RAM, in average it took the PCA-of-HOG classifier 3 ms to recognize one digit, versus 1.45 ms per digit as required by the conventional PCA classifier.

We also 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 [6], however the improvement of recognition accuracy is achieved at the cost of much increased complexity. In this regard the proposed sparse PCA-of-HOG classifier provides an attractive alternative for real-time HWDR.

V. Conclusion

A multiclass classifier based on principal component analysis (PCA) of histogram of oriented gradient (HOG) has been proposed. By applying it to the MNIST database, the PCA-of-HOG based classifier is shown to perform HWDR considerably better than the PCA applying to raw training data.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

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

[7] D. Ismailova and W.-S. Lu, "Fast classification of handwritten digits using 2D-DCT based sparse PCA," *Proc. PacRim Conf.*, Victoria, BC, Canada, Aug. 2015.

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

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

[10] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[11] N. D. Dalal and B. Triggs, "Histogram of oriented gradients for human detection," in *Proc. CVPR*, San Diego, June 2005.

[12] 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.

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