A Feature Extraction Method for Cursive Character Recognition Using Higher-Order Singular Value Decomposition

Mohammad Reza Ameri*, Medhi Haji*, Andreas Fischer†, Dominique Ponson‡, Tien D. Bui*
* Computer Science and Software Engineering Department
Concordia University
Montréal, Québec H3G 1M8, Canada
Email: mo_amer@encs.concordia.ca, m_haji@encs.concordia.ca, bui@cse.concordia.ca
† Département de Génie Électrique
École Polytechnique de Montréal
Montréal, Québec H3T 1J4, Canada
Email: andreas.fischer@polymtl.ca
‡ IMDS Software
152 Notre-Dame Est, suite 100
Montréal, Québec H2Y 3P6, Canada
Email: dponson@imds-world.com

Abstract—The use of Higher-Order Singular Value Decomposition (HOSVD) and other tensor decomposition methods are popular in the face recognition domain, yet a direct application to handwritten character recognition has not shown promising results so far. Character recognition is commonly performed in two steps: feature extraction and classification. In this paper, we propose a feature extraction algorithm based on HOSVD which is then combined with standard statistical classification. The algorithm constructs a tensor from the training data and applies HOSVD in order to obtain a feature extractor matrix for arbitrary character images. We evaluate the proposed handwriting features in combination with SVM classification for character recognition on the CEDAR benchmark data set. The results indicate that our proposed approach significantly outperforms the standard HOSVD classification method.

Index Terms—Feature evaluation and selection, optical character recognition, tensor decompositions, higher-order singular value decomposition (HOSVD).

I. INTRODUCTION

Recognition of handwritten characters using computers has been one of the first and most successful applications of pattern recognition. Optical Character Recognition (OCR) has been an active field of research for more than three decades. Hundreds of approaches have been proposed for the recognition of handwritten characters for different scripts [1]. For machine-printed Latin scripts, character recognition methods achieve very high recognition rates, at least when the level of noise is low [2]. When clear imaging is available, typical recognition rates for machine-printed characters exceed 99%. However, OCR is prone to errors when dealing with handwritten characters. Commercial applications with near-perfect recognition accuracy are only available for restricted tasks such as bank check reading [3]. In the general case, the problem is still considered as widely unsolved [4].

The difficulty of recognizing handwritten characters lies in the fact that there can be as many handwriting styles as there are people. In fact, it is widely believed that each individual’s handwriting is unique to themselves. In the discipline of forensic science, handwriting identification and verification are based on the principle that the writings of no two people are exactly alike. This means that the number of shapes that a handwritten character can take is very large, which is challenging for pattern recognition. Fig. 1 shows some examples of letters from the NIST SD19 database [5] which may be mistaken for ‘a’ without context. In a recent study, we have shown that there are at least 29 pairs of letters that may have almost identical shapes in cursive Latin handwriting [6]. Our motivation for using HOSVD for character recognition is inspired by the success of this method for face recognition [7]. Shapes of faces can also be very similar, making it necessary to distinguish different faces based on details in the image.

Handwritten character recognition is commonly performed in two steps: feature extraction and classification. Feature extraction is a crucial first step that determines how well the different characters can be distinguished in the respective feature space. For an early survey, we refer to [8]. Examples of state-of-the-art feature sets include wavelet-based representations of low-quality printed characters as well as handwritten characters [9]–[11].

In general, it cannot be predicted which feature set performs best for a specific recognition task. Yet, there are methods for feature space transformation that are applicable to any feature set and may be able to improve the class separability.
Exemplary applications for cursive handwriting include the use of Principle Component Analysis (PCA) and Independent Component Analysis (ICA) [12] as well as non-linear kernel PCA [13].

Recent advances in image representation include the development of sparse representations, which have proven successful for various applications in computer vision and pattern recognition [14]. The underlying idea is to describe an image with respect to a linear combination of representative samples such that only few coefficients of the linear combination are non-zero. Unlike PCA, the goal is not to create a feature space with a small orthogonal basis but instead use an extensive dictionary of representative samples to create an overcomplete basis. Following this procedure, semantic information like class membership can be directly propagated from the non-zero coefficients.

Higher-Order Singular Value Decomposition (HOSVD) is a promising tensor decomposition method for sparse representation. It has a high visibility in the domain of face recognition, for instance for the recognition of facial expressions [7]. A successful application to handwriting recognition was recently reported in [15] for the ten-class problem of digit classification. Promising accuracies are reported with respect to an efficient implementation, which includes a reduction of the training set, and the standard HOSVD classification method, which selects the class with the highest similarity directly from the sparse representation.

In this paper, we investigate the application of HOSVD to the more challenging problem of handwritten character recognition that considers a larger number of classes when compared with digit recognition. An experimental evaluation on the CEDAR benchmark database [16] shows that the achieved recognition accuracy is surprisingly low when compared with other state-of-the-art methods. In order to improve these initial results, we propose a different recognition approach that considers the sparse HOSVD representation as a feature vector, which is then used for statistical classification. In combination with Support Vector Machine classification using a Radial Basis Function kernel (RBF SVM), we demonstrate that the proposed approach significantly outperforms standard HOSVD classification [15], which is designed to perform character recognition as well as recognition with HOSVD. In contrast, we use HOSVD only for feature extraction. Recognition is then performed by the SVM classifier.

The remainder of this paper is organized as follows. First, the SVD and HOSVD terminology is introduced in Section II. Then, we present the proposed HOSVD-based feature extraction algorithm in Section III and provide experimental results in Section IV. Finally, conclusions are drawn in Section V.

II. Higher-Order SVD

Higher-order Singular Value Decomposition (HOSVD) is obtained by extending the concept of SVD for tensors. SVD for arbitrary matrix $A$ is defined by

$$A = U \Sigma V^T$$

where $U$ is the an orthogonal matrix containing the eigenvectors of $AA^T$ in its columns, $V$ is an orthogonal matrix containing the eigenvectors of $A^T A$ and $\Sigma$ is a diagonal matrix containing the singular values of $A$. Singular values are sorted by decreasing order in $\Sigma$. The number of singular values of $A$ is equal to $\text{rank}(A)$. Let $\hat{A}$ be an approximation of $A$ with lower rank than $A$. To build such a matrix, suppose $\text{rank}(\hat{A}) = n$ and $\text{rank}(A) = k$, $k < n$, it can be obtained by reconstructing $A$ with the $k$ first singular values of $\Sigma$.

Tensors are represented by their dimension $N$, denoted by $N$-tensor. If we consider vectors as first order tensors, $N = 1$ and matrices as second order tensors, $N = 2$ then third order tensors are three dimensional data with $N = 3$. HOSVD is a generalized concept of SVD for tensors. Unfolding or flattening of tensors [15] is the first step toward HOSVD. It involves the conversion of tensor to matrix form.

For a 3-tensor $A \in \mathbb{R}^{I \times J \times K}$ unfolding is done for each dimension. One possible flattening for 3-tensor could be:

$$\mathbb{R}^{I \times JK} \ni A(1) : a_{ijk} = a_{ij}^{(1)} \quad (v = j + (k-1)K)$$
$$\mathbb{R}^{J \times IK} \ni A(2) : a_{ijk} = a_{ij}^{(2)} \quad (v = k + (i-1)J)$$
$$\mathbb{R}^{K \times IJ} \ni A(3) : a_{ijk} = a_{ij}^{(3)} \quad (v = i + (j-1)J)$$

The second issue to deal with is matrix-tensor multiplication. For matrix $F \in \mathbb{R}^{J_n \times I_n}$ and tensor $A \in \mathbb{R}^{I_1 \times \ldots \times I_N}$, such a multiplication is called $n$-mode tensor-matrix multiplication by:

$$(A \times_n F)(i_1, \ldots, i_{n-1}, j_n, i_{n+1}, \ldots, i_N) = \sum_{i_n=1}^{I_n} (A(i_1, \ldots, i_N)F(j_n, i_n))$$
and the following property holds for tensor \( A \in \mathbb{R}^{I \times J \times K} \) and matrices \( F \in \mathbb{R}^{L \times I} \), \( G \in \mathbb{R}^{M \times J} \) and \( H \in \mathbb{R}^{N \times L} \).

\[
(A \times_1 F) \times_2 G = (A \times_2 G) \times_1 F = A \times_1 F \times_2 G \in \mathbb{R}^{L \times M \times K}.
\]

The SVD theorem states that for any matrix \( F \in \mathbb{R}^{M \times N} \), it can be written as

\[
F = U \Sigma V^T.
\]

Considering the matrix as a second order tensor this can be expressed in terms of \( n \)-mode tensor-matrix multiplication as

\[
F = \sum_{k \times 1} U \times_2 V
\]

and for a third order tensor it would be

\[
A = \varphi_1 \times_1 U \times_2 V \times_3 W.
\]

where \( A \in \mathbb{R}^{I \times J \times K} \), \( U \in \mathbb{R}^{I \times I} \), \( V \in \mathbb{R}^{J \times J} \) and \( W \in \mathbb{R}^{K \times K} \) are orthogonal matrices and \( \varphi \in \mathbb{R}^{I \times J \times K} \) is an orthogonal tensor. To compute matrices \( U, V \) and \( W \) in Eq. 1, matrices \( A(1), A(2), A(3) \) must be decomposed by SVD:

\[
\begin{align*}
A(1) &= US(1)(V(1))^T, \\
A(2) &= VS(2)(V(2))^T, \\
A(3) &= WS(3)(V(3))^T
\end{align*}
\]

and the tensor \( \varphi \) can be obtained by multiplying tensor \( A \) with \( U^T, V^T \) and \( W^T \) in 1-mode, 2-mode and 3-mode respectively:

\[
\varphi = A \times_1 U^T \times_2 V^T \times_3 W^T.
\]

Tensor \( \varphi \) and matrices \( U, V \) and \( W \) from Eq.1 can be used to compute an approximation of \( A \), denoted by \( \hat{A} \). A \((k_1, k_2, k_3)\)-rank approximation for \( A \) is obtained by keeping the first \( k_1 \), \( k_2 \), \( k_3 \) columns of \( U \), \( V \) and \( W \) respectively to produce \( U' \in \mathbb{R}^{I \times k_1} \), \( V' \in \mathbb{R}^{J \times k_2} \), \( W' \in \mathbb{R}^{K \times k_3} \), with a sub tensor \( \varphi' \in \mathbb{R}^{k_1 \times k_2 \times k_3} \), which contains the first \( k_1 \), \( k_2 \), \( k_3 \) elements of \( \varphi \). This procedure results in the tensor approximation

\[
\hat{A} = \varphi' \times_1 U' \times_2 V' \times_3 W'.
\]

Eq. 1 and Eq. 2 are key elements of the proposed feature extraction algorithm in Section III. More information about HOSVD and tensor decomposition can be found in [15] and [17].

III. FEATURE EXTRACTION

Feature selection is one of the most important steps in pattern and character recognition because it affects the performance in terms of recognition and resource usage. Our feature extraction method is based on the method of HOSVD similar to [15]. A compression measure \( p \) controls the size of the feature vector. Smaller values of \( p \) result in more tensor compression and lower resource usage.

Our feature extraction method is described as follows. Consider a training data set consisting of \( M \) sample images of \( K \) classes (\( K = 10 \) for numerals and 26 for alphabetical letters). Let \( J_i \) denote the number of sample images in class \( i \), \( i = 1, \ldots, K \), hence \( M = \sum_{i=1}^{K} J_i \). For each image of size \( m \times n \) in the data set we resize it to a \( 20 \times 20 \) matrix by using the bi-cubic approximation. We then convert each \( 20 \times 20 \) matrix representing an image into a vector of size 400 by concatenating the columns. The first step of our method is to construct a tensor \( A \) to represent the data set described above. The tensor \( A \) is a 3-tensor, \( A \in \mathbb{R}^{I \times J \times K} \), where \( I = 400 \) is the normalized size for an image, \( J = \max(J_i), \; i = 1, \ldots, K \), and \( K \) is the number of classes. Thus tensor \( A \) can be considered as a stack of \( K \) matrices \( A_i \), \( i = 1, \ldots, K \) where each matrix \( A_i \) has dimension \( 400 \times J_i \) (see Fig.2).

However, for many data sets there are different numbers of samples for different classes. Since tensor \( A \) is built from these matrices they must have the same size. To achieve this, one solution is to extend all matrices to the size of the largest \( J \), \( i = 1, \ldots, K \). The extension could be done by replicating the columns of \( A_i \). The next step is to apply HOSVD decomposition method as described in the previous section. An important step in our method to make use of low rank approximation as well as reducing computational cost is to use the approximated tensor \( \hat{A} \) instead of \( A \). \( \hat{A} \) is constructed as follows. Compression measure \( p \) determines the size of the approximated tensor \( \hat{A} \) which then produces the feature extraction matrix \( \hat{U} \) from the HOSVD method. The length of the feature vector depends on the size of \( \hat{U} \). To compute \( \hat{A} \), we use Eq. 1 to decompose \( A \) by HOSVD, and Eq. 2 where \( U' = U(:,1:p), \; V' = V(:,1:p), \; W' = W \) and tensor \( \varphi' = \varphi(1:p,1:p,:) \) where \( p < I \) and \( p < J \). Notation \((1:p,1:p,:)\) means select all rows and all columns from the first to the \( p \)-th elements. Eq. 1 is again used to decompose
In order to evaluate the proposed algorithm and to compare it with the standard HOSVD classification approach, we used the CEDAR data set [16], which contains scanned forms of cursive handwritten mail documents. The samples are split into disjoint sets for training and testing. We used the BINANUMS folder which contains alphanumeric characters. We conducted two experiments for upper case and lower case characters, respectively. The samples for training and testing are selected by the provided guideline. The upper case data contains 11,453 character for training and 1327 characters for testing. The lower case data, on the other hand, contains 7692 characters for training and 816 characters for testing. This setup has been preserved for all the experiments.

For classification we used the LIBSVM toolkit [18] to train a Support Vector Machine classifier (SVM) with Radial Basis Function kernel (RBF). A cross validation was performed on the training set over 110 pairs $(C, \gamma)$ of the SVM parameter $C$ and the RBF parameter $\gamma$. The best parameters $(C_{\text{best}}, \gamma_{\text{best}})$ were used for recognition on the test set.

The proposed combination of HOSVD-based feature extraction and RBF SVM classification is compared with the standard HOSVD classification approach, which selects the most similar class directly from the sparse HOSVD-based representation. We have verified the correctness of our implementation of the reference system, which is similar to the system proposed in [15] for digit recognition, on a ZIP code data set [19].

In Table I and Table II we report recognition accuracies for upper case and lower case experiments, respectively, and different compression measures $p \in \{16, 32, 48, 64, 128\}$. For the proposed system, results are indicated for cross validation on the training set as well as recognition on the test set. For the reference system (HOSVD classic) results are indicated for recognition on the test set.

When compared with recent benchmark results on the CEDAR database, the recognition accuracy achieved by HOSVD classic is surprisingly low. For instance, the authors of [20] report one of the best current results on this database, which is a recognition accuracy of 95.90% for upper case characters, 93.50% for lower case characters, and 94.73% for merged upper case and lower case characters of the CEDAR data set based on recursive subdivision features and RBF SVM classification [20].

Although these benchmark results cannot just yet be obtained by our proposed method, it shows significant improvements over HOSVD classic. In 9 out of 10 cases, the improvements in recognition accuracy on the test set are statistically significant (t-test, $\alpha = 0.01$). Furthermore, the proposed method achieves its best results for relatively low values of $p$. That is, a high compression and therefore a low resource usage can be achieved.
V. Conclusion

In this paper, we have investigated the application of HOSVD-based classification for handwritten character recognition. When compared with an earlier work on digit classification [15], a larger number of classes is taken into account and a surprisingly low recognition accuracy is reported on the CEDAR benchmark database.

In order to improve these initial results, we propose a different recognition approach that considers the sparse HOSVD representation as a feature vector, which is then used for statistical classification. In combination with RBF SVM classification, we demonstrate that the proposed approach significantly outperforms standard HOSVD classification.

There is clearly a potential for HOSVD-based character recognition, yet the recognition accuracy of the proposed approach is still below other state-of-the-art results. Future work includes the investigation of different image representations prior to HOSVD decomposition as well as a combination of the proposed HOSVD features with other statistical classifiers. Finally, an application of HOSVD to the task of keyword spotting could be a promising line of future research.

Acknowledgment

This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant numbers RGPIN 9265-2010 and CRDPJ 395169-09, and the fellowship project P300P2-151279 of the Swiss National Science Foundation.

References


