# Ensemble of Tensor Classifiers Based on the Higher-Order Singular Value Decomposition

Bogusław Cyganek

AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@agh.edu.pl

**Abstract.** In this paper we present an ensemble composed of classifiers operating with multi-dimensional data. Classification is performed in tensor spaces spanned by the basis obtained from the Higher-Order Singular Value Decomposition of the pattern tensors. These showed superior results when processing multi-dimensional data, such as sequences of images. However, multi-dimensionality leads to excessive computational requirements. The proposed method alleviates this problem, first by partitioning the input dataset, and then by feeding each partition into a separate tensor classifiers of the ensemble. Despite the computational advantages, also accuracy of the ensemble showed to be higher compared to a single classifier case. The method was tested in the context of object recognition in computer vision. In the paper we discuss also methods of input image prefiltering in order to increase accuracy. The conducted experiments show high efficacy of the proposed solution.

**Keywords:** Classification, HOSVD, ensemble of classifiers.

## 1 Introduction

Object recognition by computers requires construction of accurate and fast classifiers. There are many examples of these which are reported in literature [16][18][19]. However, many existing methods do not account for the multi-dimensionality of the classified data. Recently, this issue was addressed with help of the tensor analysis. One of the first methods which utilize this approach was face recognition system proposed by Vasilescu *et al.* [19]. In their approach tensors constitute a major mathematical tool to cope with multiple factors of face patters, which can be represented under different poses, views, illuminations, etc. Because of this, the method was called tensor-faces. Another system that is based on tensor analysis was proposed by Savas *et al.* [15]. It was applied to the problem of handwritten digits recognition [13]. The method by Savas *et al.* assumes tensor decomposition which allows an equivalent representation of a tensor as a product of a core tensor and the unitary mode matrices. This decomposition is known as the Higher-Order Singular Value Decomposition (HOSVD), and is related to the Tucker decomposition [1][12][9]. A version of this method was then used by Cyganek in the system for road signs recognition [3]. In this case the input tensor is built from artificially generated

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 578–589, 2012.

deformed versions of the prototype road sign exemplars. All these systems, which are based on HOSVD, show very high accuracy and high speed of response. However, computation of the HOSVD from large size tensors is computationally demanding. The decomposition algorithm requires computation of a sequence of SVD decompositions of matrices obtained from the flattened input tensor. However, dimensions of these matrices can be huge since they are multiplications of all dimensions of the input tensor. In many applications this can be very problematic. The method presented in this paper shows how to alleviate this problem by construction of an ensemble of tensor-based classifiers. In the proposed solution tensors are of much smaller size than in a case of a single tensor based classifier. As it will be shown, despite the computational advantages, the proposed ensemble shows also superior accuracy when compared with a single classifier.

The method was tested in the task of handwritten digits recognition and showed good results and high speed of operation. Detailed experimental results are provided for the highly demanding USPS dataset [6][20]. These verified our assumption on high accuracy and lower computational complexity of the proposed ensemble.

The rest of the paper is organized as follows: Section 2 presents basics on N-Mode Principal Component Analysis for pattern recognition. In Section 3 we provide details of the proposed methods, i.e. construction of the ensemble of HOSVD classifiers. Experimental results with discussion of the obtained scores are provided in Section 4. The paper ends with conclusions in Section 5.

## 2 N-Mode Principal Component Analysis for Pattern Recognition

Tensors in data mining and classification are defined as multidimensional arrays. They generalize such concepts as scalars, vectors, and matrices, which all are tensors. Processing and analysis of images builds well into this framework due to a multi-dimensional nature of data in video streams. However, an analysis of contents of a video represented as tensors requires their proper decomposition. There are many tensor decompositions which allow either their analysis or compact representation. However, one of the most popular is the already mentioned HOSVD [1][11][9]. As it will be shown in the next sections, HOSVD can be used to build orthogonal spaces which can be then used for pattern recognition in a similar way to the standard PCA based classifiers [4][18]. However, before we show properties and the algorithm of computation of the HOSVD, we present briefly some of the most important concepts of the tensor algebra.

### 2.1 Tensor Algebra Concepts

Although tensors can be multi-dimensional, in many methods it is convenient to represent them in the matrix-like, or flattened, form. More specifically, for a *P-th* order tensor $\mathcal{T} \in \Re^{N_1 \times N_2 \times \ldots N_P}$, the *k-mode vector* of $\mathcal{T}$ is defined as a vector obtained from the elements of $\mathcal{T}$ by varying only one index $n_k$, while keeping all other fixed. Let us assume that from the tensor $\mathcal{T}$ the following matrix

$$\mathbf{T}_{(k)} \in \Re^{N_k \times (N_1 N_2 \ldots N_{k-1} N_{k+1} \ldots N_P)} \tag{1}$$

is formed. Now columns of $\mathbf{T}_{(k)}$ are *k-mode* vectors of $\mathcal{T}$. The $k$-mode representation of a tensor is obtained by selecting the $k$-th index which becomes a row index of its flatten representation. On the other hand its column index is a product of all other $P$-1 indices. Nevertheless, where an element of the tensor is stored in memory depends on the chosen permutation of these $P$-1 indices, which results in ($P$-1)! possibilities. From these only two, i.e. forward and backward cycle modes, are used [11].

The second key concept is a $k$-mode multiplication of a tensor $\mathcal{T} \in \Re^{N_1 \times N_2 \times \ldots N_P}$ and a matrix $\mathbf{M} \in \Re^{Q \times N_k}$. In result of such a multiplication a tensor $\mathcal{S} \in \Re^{N_1 \times N_2 \times \ldots N_{k-1} \times Q \times N_{k+1} \times \ldots N_P}$ is obtained, which elements can be expressed as follows

$$\mathcal{S}_{n_1 n_2 \ldots n_{k-1} q n_{k+1} \ldots n_P} = \left(\mathcal{T} \times_k \mathbf{M}\right)_{n_1 n_2 \ldots n_{k-1} q n_{k+1} \ldots n_P} = \sum_{n_k=1}^{N_k} t_{n_1 n_2 \ldots n_{k-1} n_k n_{k+1} \ldots n_P} m_{q n_k}. \tag{2}$$

Finally, to analyze contents of a tensor a proper decomposition needs to be applied. The HOSVD decomposition, used also in the proposed method, allows any $P$-dimensional tensor $\mathcal{T} \in \Re^{N_1 \times N_2 \times \ldots N_m \times \ldots N_n \times \ldots N_P}$ to be equivalently represented in the following form [11][12]

$$\mathcal{T} = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \ldots \times_P \mathbf{S}_P . \tag{3}$$

In the above, $\mathbf{S}_k$ are unitary matrices of dimensions $N_k \times N_k$, called *mode matrices*. $\mathcal{Z} \in \Re^{N_1 \times N_2 \times \ldots N_m \times \ldots N_n \times \ldots N_P}$ is *a core tensor* which fulfills the following properties [11][12]:

1. Two subtensors $\mathcal{Z}_{n_k=a}$ and $\mathcal{Z}_{n_k=b}$, are orthogonal for all possible values of $k$ for which $a \neq b$, i.e.

$$\mathcal{Z}_{n_k=a} \cdot \mathcal{Z}_{n_k=b} = 0 , \tag{4}$$

2. All subtensors of $\mathcal{Z}$ for all $k$ can be ordered according to their Frobenius norms

$$\left\| \mathcal{Z}_{n_k=1} \right\| \geq \left\| \mathcal{Z}_{n_k=2} \right\| \geq \ldots \geq \left\| \mathcal{Z}_{n_k=N_P} \right\| \geq 0 , \tag{5}$$

Finally, the *a-mode* singular value of $\mathcal{T}$ is defined as follows

$$\left\| \mathcal{Z}_{n_k=a} \right\| = \sigma_a^k . \tag{6}$$

In the next section we discuss an algorithm for computation of the HOSVD.

### 2.2 Computation of the Higher-Order Singular Value Decomposition

Lathauwer proposed a method of computation of the HOSVD which is based on successive application of the SVD decompositions to the flattened matrices of a given tensor [11]. Thus, HOSVD of a $P$-dimensional tensor $\mathcal{T}$ is presented by the following algorithm [11][12]:

**1.** For each $k$=1, …, $P$ do:
  **a.** Flatten tensor $\mathcal{T}$ to obtain $\mathbf{T}_k$, from Eg.(1)
  **b.** Compute $\mathbf{S}_k$ from the SVD decomposition of $\mathbf{T}_k$

$$\mathbf{T}_k = \mathbf{S}_k \mathbf{V}_k \mathbf{D}_k^T \tag{7}$$

**2.** Using all $\mathbf{S}_k$ compute the core tensor:

$$\mathcal{Z} = \mathcal{T} \times_1 \mathbf{S}_1^T \times_2 \mathbf{S}_2^T \ldots \times_P \mathbf{S}_P^T \tag{8}$$

**Fig. 1.** An algorithm for computation of the HOSVD of tensors

Because $\mathbf{S}_k$ are orthogonal, the core tensor $\mathcal{Z}$ can be expressed as

$$\mathbf{Z}_{(k)} = \mathbf{S}_k^T \mathbf{T}_{(k)} \left[ \mathbf{S}_{k+1} \otimes \mathbf{S}_{k+2} \otimes \ldots \otimes \mathbf{S}_P \otimes \mathbf{S}_1 \otimes \mathbf{S}_2 \otimes \ldots \otimes \mathbf{S}_{k-1} \right]. \tag{9}$$

From the algorithm in Fig. 1 we see that computation of the HOSVD requires a sequence of SVD decomposition of matrices obtained from the input tensor. However, dimensions of these matrices can be very large since they are just multiplications of all dimensions of the decomposed tensor, as shown in Eg. (1). In many practical cases this poses a real problem. The method presented in this paper shows how to overcome this problem with construction of an ensemble of tensors, which are of lower sizes, however. As will be shown, apart from this computational advantage, such ensemble of classifiers shows also better accuracy when compared with a single classifier.

### 2.3 Pattern Recognition in the Tensor Spanned Spaces

For each mode matrix $\mathbf{S}_i$ in (3) the following sum can be constructed

$$\mathcal{T} = \sum_{h=1}^{N_P} \mathcal{T}_h \times_P \mathbf{s}_P^h, \tag{10}$$

thanks to the commutative properties of the $k$-mode multiplication. In the above

$$\mathcal{T}_h = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \ldots \times_{P-1} \mathbf{S}_{P-1} \tag{11}$$

constitute the basis tensors and $\mathbf{s}^h_P$ are columns of the unitary matrix $\mathbf{S}_P$. Because $\mathcal{T}_h$ is of dimension $P$-1 then $\times_P$ in (10) is an outer product, i.e. a product of two tensors of dimensions $P$-1 and 1. Moreover, due to the orthogonality properties of the core tensor $\mathcal{Z}$ in (11), $\mathcal{T}_h$ are also orthogonal. Thus, they can constitute a basis which spans a sub-space. This property is used to construct a HOSVD based classifier, as follows.

In the tensor space spanned by $\mathcal{T}_h$, pattern recognition can be stated as testing a distance of a given test pattern $\mathbf{P}_x$ to its projections in each of the spaces spanned by the set of the bases $\mathcal{T}_h$ in (11). This can be expressed as the following minimization problem [15]

$$\min_{i,c_h^i} \underbrace{\left\| \mathbf{P}_x - \sum_{h=1}^{H} c_h^i \mathcal{T}_h^i \right\|^2}_{Q_i}, \tag{12}$$

where the scalars $c^i_h$ denote unknown coordinates of $\mathbf{P}_x$ in the space spanned by $\mathcal{T}_h^i$, and $H \leq N_P$ denotes a number of chosen dominating components.

To solve (12) the squared norm $Q$ of (12) is created for a selected $i$. Assuming further that $\mathcal{T}_h^i$ and $\mathbf{P}_x$ are normalized the following is obtained (the *hat* indicates normalized tensors)

$$\rho_i = 1 - \sum_{h=1}^{H} \left\langle \hat{\mathcal{T}}_h^i, \hat{\mathbf{P}}_x \right\rangle^2 . \tag{13}$$

Thus, to minimize (12) we need to maximize the following value

$$\hat{\rho}_i = \sum_{h=1}^{H} \left\langle \hat{\mathcal{T}}_h^i, \hat{P}_x \right\rangle^2 , \tag{14}$$

In other words, the HOSVD based classifier returns a class $i$ for which its $\rho_i$ from (14) is the largest.

## 3 Construction of the Ensemble of HOSVD Classifiers

It was shown that an ensemble of even simple classifiers can perform better that a complex but single classifier [10][14][7]. As we will show, this holds also for the HOSVD based classifiers. Apart from the higher overall accuracy, an ensemble of HOSVD classifiers offers also computational advantages such as lower memory requirements due to reduced training data partitions, as well as the parallel run-time structure.

To build an ensemble of HOSVD based classifiers we propose to use bagging and image preprocessing. Bagging consists in creating a number of variants $\mathbf{X}_T^{(i)}$ of the training set $\mathbf{X}_T$, by a uniform data sampling from $\mathbf{X}_T$ with replacement (a bootstrap aggregation method) [17]. It was shown that bagging can reduce variance of a classifier and improve its generalization properties [5]. Each data variant is used to train a separate member of the ensemble, which in our case is the HOSVD classifier. A number of data in each variant $\mathbf{X}_T^{(i)}$ is one of the training parameters of the ensemble. However, the issue is that in the case of an ensemble of classifiers, each classifier can be trained with data variant which is less numerous than the maximally available number of training points and the overall accuracy of the ensemble will be still higher that for a single classifier case. Thanks to this strategy we can cope easily with massive data, training one classifier at a time. Additionally, it is possible to extend the ensemble with new members if new training data are obtained in the later time. An analysis of the experimental results confirms these suppositions.

Additionally, to improve performance of the classifiers we propose to add image prefiltering stage, as shown in Fig. 2. In our experiments we tested the following prefilters:

1. Affine image warping (bilinear interpolation);
2. Edge detection with the Savitzky-Golay filters;
3. Phase component of the structural tensor;
4. Census transform;
5. Gaussian noise addition.

Detailed algorithms with code examples of the above procedures are described in [2].

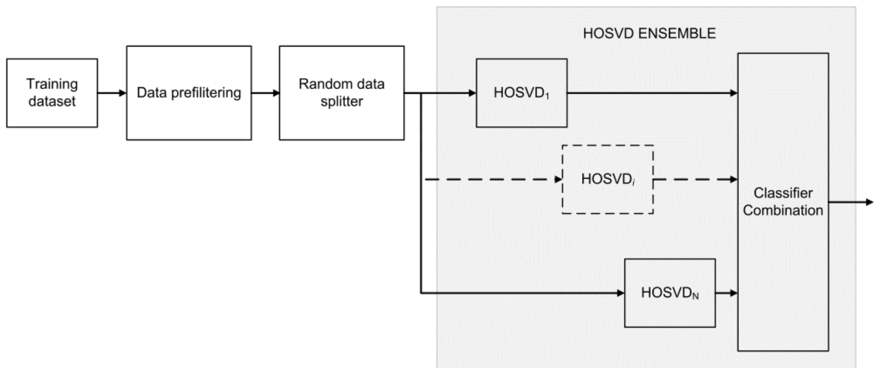


**Fig. 2.** Data flow in the proposed construction of the HOSVD ensemble of classifiers

Fig. 2 shows the construction steps (white blocks), as well as structure of the HOSVD ensemble (gray blocks). Each of the individual HOSVD$_i$ experts in Fig. 2 is trained to recognize exactly ten classes of the handwritten digits, however each is fed with different dataset. That is, each HOSVD$_i$ is trained with its specific data partition obtained by random sampling a number of training data (i.e. images of the handwritten digits) from the set of all data (bagging). This is done in the random splitter block in Fig. 2, using a random generator with a uniform distribution.

Summarizing, the training parameters of the entire ensemble are as follows:

1. Number of member classifiers in the ensemble;
2. Size of data partitions in the bagging process;
3. Type of the preprocessing filter;
4. Number of components $H$ considered in (14);

In the run-time, answers of each of the classifiers are combined with the majority voting scheme [8][10]. Summarizing, the proposed ensemble of HOSVD classifiers offers the following advantages:

- Significant reduction of memory requirements during training;
- Reduction of computations (due to lower size of the matrices);
- Possible incremental build (e.g. if new training samples are coming at later time, they can compose a new member of the ensemble);
- Proper data prefiltering can result in up to 1-2% improvement (however, for different datasets, this needs experimental verification);
- The method naturally leads to the parallel training and run-time architectures;

Certainly, the most important factor is the overall accuracy of an ensemble. As we show in the next section, this and also all of the aforementioned postulates were verified experimentally.

## 4 Experimental Results

The presented method was entirely implemented in C++, supported by the HIL library [2]. The experiments were run on the computer with 8 GB RAM and with Pentium® Quad Core Q 820 (clock 1.73 GHz). For the experiments the USPS dataset was used [6][20]. The same set was also used by Savas *et al.* [15]. This dataset contains selected and preprocessed scans of the handwritten digits from envelopes by the U.S. Postal Service. Fig. 3 depicts exemplars of each digit from the training set (Fig. 3a), and from the testing set (Fig. 3b), respectively.

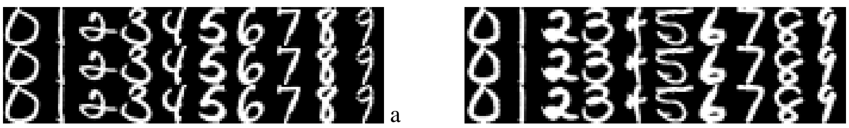
a b

**Fig. 3.** Visualization of the two data sets from the ZIP database (from data). Ten exemplars of each digit from the training set (a), and from the testing set (b).

Each test and train pattern originally is in a form of a 16×16 gray level image. Since this dataset is perceived as a relatively difficult for machine classification (reported human error is 2.5%), it has been used for comparison of different classifiers [13][15]. The database is divided into the training and testing partitions, counting 7291 and 2007 exemplars, respectively. Detailed numbers of training and test patterns for each digit are contained in Table 1.s

Each experimental setup was run number of times (from 3 to 10, depending on computational complexity) and an average answer is reported. In all cases the Gaussian noise was added to the input image at level of 10%, in accordance with the procedure described in [2].

In the first group of experiments we tested influence of the number of data points used in bagging process on accuracy of the ensembles with different number of members. Fig. 4 shows obtained results of these tests in respect to the number of classifiers in the ensembles. We see that the best accuracies were obtained for partitions of 560 points. However, almost similar accuracies are for partitions counting 256 points. It is interesting to observe differences of accuracy for a single classifier system (i.e. one classifier) and ensembles even with only 2 or 3 members. In this case accuracy increases rapidly by about 1%, regardless of a number of data points used in bagging. This acknowledges our assumption of better accuracy in the case of the ensembles with diversity obtained with data bagging. At the same time, each HOSVD classifier of an ensemble requires much less memory during training since it deals with a smaller training set.

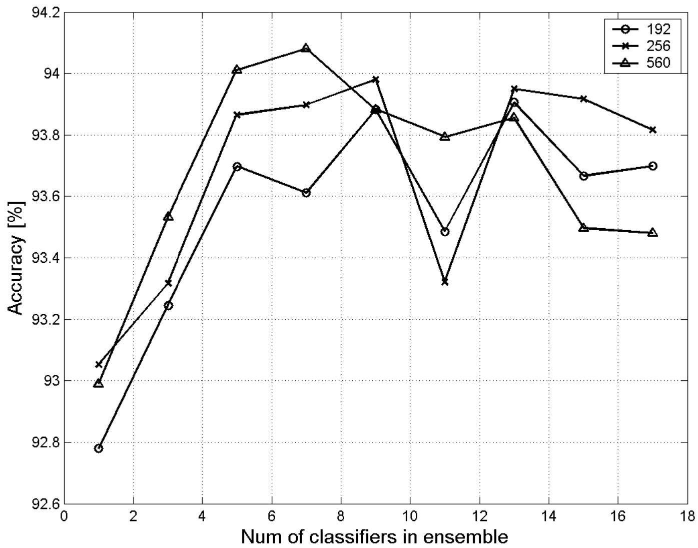


**Fig. 4.** Accuracy vs. number of classifiers in the ensemble for three sizes of data samples in the bagging process (192, 256, 560). Input images of size 16×16. In Eq. (14) number of components $H$=16.

In the next experiments we compared performance of the ensembles of different number of members in respect to the two different sizes of the input images: 16×16 vs. 32×32 pixels. Fig. 5 depicts obtained accuracies in these experiments. It is evident that images of larger size always result in higher accuracy. Also a difference between a single classifier and an ensemble of two or more members is not so rapid in the case of larger input images. The other tests with other of the aforementioned prefilters which changed signal representation such as, computing edges, phases of the structural tensor, as well as computing the Census measure did not produce better accuracies than bare intensity signal. However, as it was already pointed out, the interpolation to larger size increases accuracy. Unfortunately, this is burdened with higher computational costs. Thus, in our setup a trade-off was set to the resolution of 32×32 pixels. Larger images did not produce much better accuracy, whereas computational demands grew excessively.

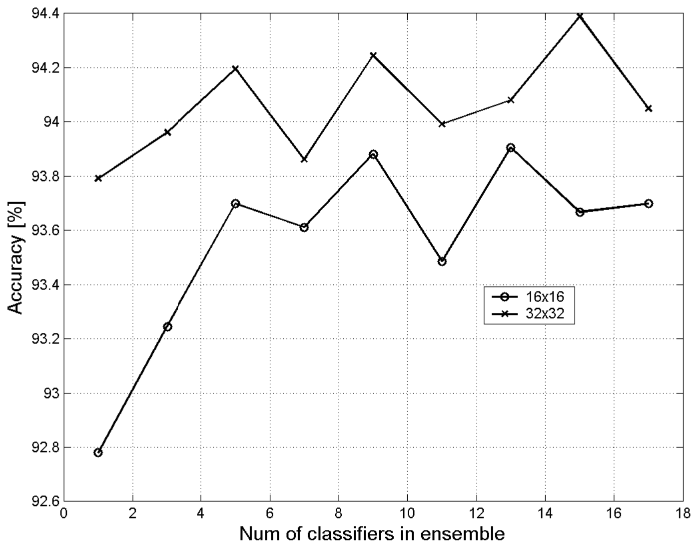


**Fig. 5.** Accuracy vs. number of classifiers in the ensemble for two different sizes of input images: original (16×16) vs. enlarged by image warping (32×32). In both cases 192 data samples (images) were used in bagging. In Eq. (14) number of used components $H$=16.

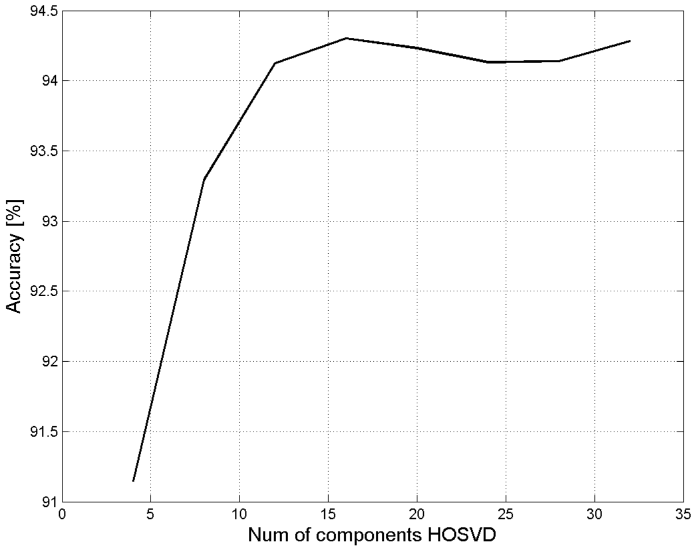


**Fig. 6.** Accuracy vs. number of components $H$ in Eq. (14). Input images were warped to 32×32, number of classifiers in ensemble set to 15, bagging partitions of 192 images used.

In the next experiment we tried to figure out the optimal value of the number of components $H$ of the HOSVD approximation in Eq. (14). Fig. 6 depicts accuracy vs. parameter $H$. In this case a setup with best accuracy was chosen (see Fig. 5). That is, the input images were warped to 32×32 pixels, number of classifiers in the ensemble was set to 15, bagging partitions of 192 images. It is evident that there is an optimal value of $H$=16. Beyond that point, when $H$ is increased, accuracy reaches its plateau. The reason of this is that increasing $H$ we consider components usually associated with noise (like in the PCA for matrices). Further experiments with different settings revealed very similar value of the parameter $H$. The reported value of $H$ by Savas *et al.* is 12 [15]. However, in our setups we deal with ensembles, so the values of $H$ can differ slightly.

Table 1 shows numbers of training and test patterns in the dataset. It shows also detailed accuracies for each digit separately for the ensemble which reached the best accuracy in Fig. 5. This ensemble counts 15 members, input data were transformed to 32×32 resolution, 192 training images were used, and number of components in (14) is $H$=16.

**Table 1.** Accuracies and parameters for each digit separately. Experimental setup: 15 members in the ensemble, input data transformed to 32×32 resolution, 192 training images from bagging, $H$=16 components.

| Digit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| #Train | 1194 | 1005 | 731 | 658 | 652 | 556 | 664 | 645 | 542 | 644 |
| #Test | 359 | 264 | 198 | 166 | 200 | 160 | 170 | 147 | 166 | 177 |
| Accur. | 0.986 | 0.981 | 0.899 | 0.880 | 0.930 | 0.925 | 0.982 | 0.959 | 0.910 | 0.960 |

Results in Table 1 show that the worse detailed accuracy was obtained for digit "3". Indeed it can be confused with "8", for which obtained accuracy is also not the best one. The best particular accuracies were obtained for "0", "1", and "6". Measured average classification time for a single data point is in order of 1-2 ms.

## 5 Conclusions

In this paper we propose a novel method of construction of the ensemble of tensor classifiers based on the Higher-Order Singular Value Decomposition. It was shown that although the HOSVD based classifier allows high accuracy and speed of classification, its construction for large training sets can be computationally demanding. The proposed method alleviates this problem by construction of the ensemble of the HOSVD classifiers and data bagging technique. Thanks to this, the tensors for the HOSVD decompositions are of lower size than the original tensor which would contain all training data. Moreover, the constructed ensemble of classifiers achieves higher overall accuracy as compared to a solution with the single classifier. The method was tested in the context of handwritten digits recognition. In the paper we discussed also methods of prefiltering of the input patterns in order to increase accuracy of the system. In this respect we checked affine warping of images, addition of noise, Census transformation, as well as the structural tensor. We showed that application of the first of the mentioned filters can increase accuracy of the system. The obtained results acknowledged our assumptions.

**Acknowledgement.** The work was supported in the years 2011-2012 from the funds of the Polish National Science Centre NCN, contract no. DEC-2011/01/B/ST6/01994.

## References

1. Cichocki, A., Zdunek, R., Amari, S.: Nonnegative Matrix and Tensor Factorization. IEEE Signal Processing Magazine 25(1), 142–145 (2008)
2. Cyganek, B., Siebert, J.P.: An Introduction to 3D Computer Vision Techniques and Algorithms. Wiley (2009)
3. Cyganek, B.: An Analysis of the Road Signs Classification Based on the Higher-Order Singular Value Decomposition of the Deformable Pattern Tensors. In: Blanc-Talon, J., Bone, D., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2010, Part II. LNCS, vol. 6475, pp. 191–202. Springer, Heidelberg (2010)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley (2001)
5. Grandvalet, Y.: Bagging equalizes influence. Machine Learning 55, 251–270 (2004)
6. Hull, J.: A database for handwritten text recognition research. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(5), 550–554 (1994)
7. Jackowski, K., Woźniak, M.: Algorithm of designing compound recognition system on the basis of combining classifiers with simultaneous splitting feature space into competence areas. Pattern Analysis and Applications 12, 415–425 (2009)
8. Kittler, J., Hatef, M., Duing, R.P.W., Matas, J.: On Combining Classifiers. IEEE PAMI 20(3), 226–239 (1998)

9. Kolda, T.G., Bader, B.W.: Tensor Decompositions and Applications. SIAM Review, 455–500 (2008)
10. Kuncheva, L.I.: Combining Pattern Classifiers. Methods and Algorithms. Wiley Interscience (2005)
11. de Lathauwer, L.: Signal Processing Based on Multilinear Algebra. PhD dissertation, Katholieke Universiteit Leuven (1997)
12. de Lathauwer, L., Moor de, B., Vandewalle, J.: A Multilinear Singular Value Decomposition. SIAM Journal of Matrix Analysis and Applications 21(4), 1253–1278 (2000)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. Proc. IEEE on Speech & Image Processing 86(11), 2278–2324 (1998)
14. Polikar, R.: Ensemble Based Systems in Decision Making. IEEE Circuits and Systems Magazine, 21–45 (2006)
15. Savas, B., Eldén, L.: Handwritten digit classification using higher order singular value decomposition. Pattern Recognition 40, 993–1003 (2007)
16. Szeliski, R.: Computer Vision. Algorithms and Applications. Springer, Heidelberg (2011)
17. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Academic Press (2009)
18. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3(1), 71–86 (1991)
19. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear Analysis of Image Ensembles: TensorFaces. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 447–460. Springer, Heidelberg (2002)
20. `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`