Transferring Neural Representations for Low-Dimensional Indexing of Maya Hieroglyphic Art

Edgar Roman-Rangel[1](✉), Gulcan Can[2,3], Stephane Marchand-Maillet[1],
Rui Hu[2,3], Carlos Pallán Gayol[5], Guido Krempel[6], Jakub Spotak[4],
Jean-Marc Odobez[2,3], and Daniel Gatica-Perez[2,3]

[1] Department of Computer Science, University of Geneva, Geneva, Switzerland
{edgar.romanrangel,stephane.marchand-maillet}@unige.ch
[2] Idiap Research Institute, Martigny, Switzerland
{gcan,rhu,odobez,gatica}@idiap.ch
[3] École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[4] Comenius University, Bratislava, Slovakia
spotak.jakub@gmail.com
[5] University of Bonn, Bonn, Germany
pallan.carlos@gmail.com
[6] Bonn, Germany
tilalhix@googlemail.com

Abstract. We analyze the performance of deep neural architectures for extracting shape representations of binary images, and for generating low-dimensional representations of them. In particular, we focus on indexing binary images exhibiting compounds of Maya hieroglyphic signs, referred to as glyph-blocks, which constitute a very challenging dataset of arts given their visual complexity and large stylistic variety. More precisely, we demonstrate empirically that intermediate outputs of convolutional neural networks can be used as representations for complex shapes, even when their parameters are trained on gray-scale images, and that these representations can be more robust than traditional handcrafted features. We also show that it is possible to compress such representations up to only three dimensions without harming much of their discriminative structure, such that effective visualization of Maya hieroglyphs can be rendered for subsequent epigraphic analysis.

Keywords: Shape retrieval · Neural networks · Dimensionality reduction

1 Introduction

Deep Learning has become the standard technique to face many problems in visual recognition [12], where its potential for dealing with shape images has mainly focused on recognizing numeral instances [7], generic shapes [21], and 3D

G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part I, LNCS 9913, pp. 842–855, 2016.
DOI: 10.1007/978-3-319-46604-0_58

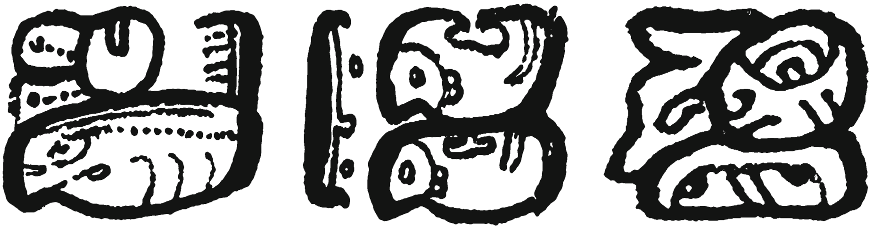

Fig. 1. Three glyph-blocks with 2, 3, and 3 individual glyph-signs, respectively: T0267, T0613; T0001, T0671, T0671; and T0115, T0667, T0024.

shapes [22]. However, more challenging scenarios remain to be explored, like the case of ancient inscriptions [8].

Such is the case of the ancient Mayan languages, which were recorded by means of a highly sophisticated system of hieroglyphic writing, comprising several thousand hieroglyphic signs, which has left us with an exceptionally rich artistic legacy. Maya hieroglyphs constitute a collection of signs highly rich in terms of style as reflected by their intricate visual structures and variations. Therefore, enabling effective retrieval of visually similar hieroglyphs can help epigraphers understand the structure of ancient languages and scribal practices. Also, given a visual language model, it can help them recognize ambiguous instances. However, these are very challenging tasks because of the highly visual complexity of the hieroglyphic signs, including visual variations of them. Figure 1 shows three examples of Maya hieroglyphs.

Following the successful trend of deep learning to analyze shapes [21,22], we propose: (1) to index shapes of Maya hieroglyphs by using representations extracted from intermediate layers of Convolutional Neural Networks (CNN), namely the Vgg-m network [19]; and (2) to use advanced dimensionality reduction methods for enabling effective visualization of them. In particular, we focus on binary images containing glyph-blocks from the ancient Maya culture. This is, groups of individual hieroglyphic-signs, which are combined to form coherent sentences, and whose combinations may vary arbitrary in location and scale, according to scribal styles and practices.

More precisely, we use the output of intermediate layers of Vgg-m as the representation of the glyph-blocks. However, given the relative small size of our dataset, a constraint for effectively training the parameters of the network is imposed. Therefore, we kept the network parameters as learned from the Imagenet dataset [12] instead of training it with the shapes of Maya hieroglyphs. This approach has proven effective in previous works [16,23], and in this work we demonstrate experimentally that it can be exploited to the extreme of representing binary images with parameters learned on images of different nature, i.e., gray-scale images. We compare the CNN intermediate representations with the Histogram of Orientations Shape Context (HOOSC) [17], a handcrafted local shape descriptors which has proven robust for dealing with complex shapes. Our evaluation shows that representations extracted from intermediate layers of the Vgg-m net outperform the retrieval precision of HOOSC.

In turn, we also investigate the potential of deep learning methods for generating low-dimensional shape representations, which could allow us to visualize glyph-blocks for further epigraphic/palaeographic analysis. Namely, we use supervised autoencoders and t-SNE [19] to map our data onto very short representations [11,14]. Autoencoders have been used to learn local descriptors and found to be competitive with respect to handcrafted descriptors [3]. An early use of autoencoders for image retrieval proposed a binary representation for hashing-based retrieval, which proved to be highly effective [11]. A thorough review of representation learning techniques and details about autoencoders can be found in [1]. Our results show that these techniques provide more robust short representations with respect to traditional PCA [10]. Namely, t-SNE obtained slightly improved retrieval performance with respect to the use of autoencoders.

The remaining of this paper is organized as follows. Section 2 discusses work related to description and indexing of Maya hieroglyphs. Section 3 details our methodology. Section 4 explains the dataset we used. Section 5 presents our experimental protocol and results. And Sect. 6 lists our conclusions.

2 Related Work

Binary shapes have been previously described by using autoencoders with logistic transfer functions [7]. Specifically, the work in [7] gives a detailed description of the architecture of autoencoders, and discusses their potential for processing faces and digits with visualization purposes. However, the dataset of digits is far less challenging that the Maya hieroglyphs we process in this work.

In a related direction, shapes of generic object (i.e., manual sketches) were successfully described by using convolutional neural networks (CNN) to perform sketch-based 3D shape retrieval [21]. In particular, that work proposes a methodology for cross-modal retrieval based on the use of siamese convolutional networks, which work well for shapes of generic objects.

The VGG-m net [19] is a deep CNN proposed to address the problem of classifying large datasets of images. It has been evaluated varying its architecture and parameters, and it was shown that deep CNN are suitable for extracting visual patterns from images at different levels of abstraction. Thus, we use two of its intermediate layers to compute shape representations in this work.

Later, it has been shown that it is possible to use convolutional neural networks off-the-shelf [16]. This is, to use the parameters of a network as learned on a training dataset of different nature than the test dataset. Variations of this approach might use such parameters as initial solution and then perform a fine tuning of them on a training set of similar nature than the test dataset. We, however, use the VGG-m network parameters off-the-shelf as learned on the Imagenet dataset i.e., with no fine tuning. The reason for this is that our dataset is not large enough for conducting an adequate training of the network. Nevertheless, off-the-shelf parameters work well in practice, as shown by our results.

Regarding the processing of Maya hieroglyphs, a retrieval system encoding glyph context information was proposed in [9], where glyphs within a block were converted into a first-order Markov chain, statistical glyph co-occurrence model and shape representation were combined for glyph retrieval. HOOSC [17] descriptor with Bag-of-Words pipeline was used to represent shape feature of glyphs. The proposed system was further evaluated in [8], where two statistical glyph co-occurrence language models extracted from diverse data sources were tested on two different Maya glyph datasets extracted from codices and monuments separately.

In [3], two types of shape representations were studied in a bag-of-words based pipeline to recognize Maya glyphs. The first was a knowledge-driven HOOSC representation, and the second was a data-driven representation obtained by applying an unsupervised Sparse Autoencoder (SA). In addition to the glyph data, the generalization ability of the descriptors was investigated on the larger-scale sketch dataset [5]. From their experiments, the data-driven representation performed overall in par with hand-designed representation for similar locality sizes for which the descriptor was computed. It is also observed that a larger number of hidden units, the use of average pooling, and a larger training data size in the SA representation improved the descriptor performance. Additionally, it is noted that the characteristics of the data and stroke size played an important role in the learned representation.

A limitation of the work presented in [3] was that a single layer autoencoder was used. We expect deeper autoencoders to provide better overall shape representations. In our paper, we designed an autoencoder with 3 hidden layers. Furthermore, we use the learned hidden representations for dimensionality reduction and visualization purposes instead of using them as convolutional filters. This is why the deepest layer in our autoencoder model has only 3 units. Another different aspect of this work and our work is training separate autoencoders for each class. This brings supervision to our overall model.

3 Approach

This section explains the preprocessing steps used for description, the supervised autoencoder model and its training, and the procedures for dimensionality reduction.

3.1 Preprocessing

In this work, we face the problem of describing shapes of very high visual complexity. This problem has been faced previously using local shapes descriptors [17], which reported high success rates. Therefore, we also rely on local shape descriptors, both for the baseline method and the proposed approach.

Binarization. Following the state-of-the-art on document binarization, we first applied a robust segmentation procedure to the images. We found that the graph-based segmentation strategy [6] applied on the image filtered by mean-shift over a combination of its HSV components added with spatial localisation, provides robust segmentation results. In particular, this segmentation is robust to small noisy artifact found in the pictures. We then simply computed average gray-scale color for every region and applied a fixed threshold to obtain the binary mask.

Description. We computed robust descriptors for the glyph-blocks using 3 approaches:

- BoW: We sampled, uniformly, 15 % of points from the medial axis of the shape, and used them as points of interest on which to compute Histograms of Orientations Shape Context (HOOSC) [17]. On average, this sampling rate resulted in 804.6 ± 417.4 points per glyph-block. To produce final representations for subsequent actions, we quantized the sets of HOOSC descriptors to generate bag representations. In particular, it has been shown that a visual vocabulary of 2000 words works well for the HOOSC descriptor [17]. Therefore, we used a randomly selected set of HOOSC's to compute $k = 2000$ visual words using k-means clustering [13]. Compared to other methods for shape description, HOOSC has obtained higher retrieval results dealing with individual hieroglyphs and generic shapes [17], as well as localizing specific shapes within large images [18].
- conv5: We used the output of the fifth convolutional layer of the Vgg-m network [19] as shape representation.
- fc7: This is the last fully-connected layer of the Vgg-m network.

The VGG-m network is inspired from Zeiler and Fergus's network (ZFnet) [24] for ImageNet data. ZFnet builts upon AlexNet [12] (8-layer network with 5-convolutional and 3 fully-connected layers) with few differences: smaller stride and receptive field sizes in the first convolutional layer and larger stride in conv2 layer. As the only difference of the VGG-m network with respect to the ZFnet, the conv4 layer has half number of filters (512 vs. 1024). In both cases, conv5 and fc7, the network parameters are kept as learned from the Imagenet data [12]. We decided to use the output of these layers as they are shown to be competitive as global image descriptors, especially fc7-layer activations outperform the shallow handcrafted representationsfor many computer vision tasks [16]. As pointed out in [24], the activations from early layers learn primitive edge and color structures, and each next layer learns more complex combinations of the previous layer activations, i.e., edges, object parts, and in the end, object templates. Furthermore, the experiments in [23] show that layers towards the end of the network are more dataset-specific, whereas the output of middle layers has better generalization for different datasets.

Table 1 shows the dimensionality of each of the three shape representations previously described.

Table 1. Number of features of each of the three shape representations.

Representation	BoW	conv5	fc7
Number of features	2,000	86,528	4,096

3.2 Model Training

Following previous works for dimensionality reduction with autoencoders [7,11], we trained several 7-layer fully connected autoencoders, one per visual class in our datasets. Here, the input layer contains as many units as the dimensionality of the shape representation, i.e., 2000 for HOOSC, 86528 for conv5, and 4096 for fc7. We decided to use 3 units for the 4-th layer, which is the deepest layer of the encoding phase, as we are interested in producing output representations that are suitable for visualization purposes. The number of units in the two intermediate layers, 500 and 100 units, was chosen after trying several combinations, such that it minimized the reconstruction error. Also, we used a fully connected architecture between consecutive layers, and a Logistic function in all units of the autoencoder. Figure 2 shows the architecture of the supervised autoencoder used in this work and the definition of its units.

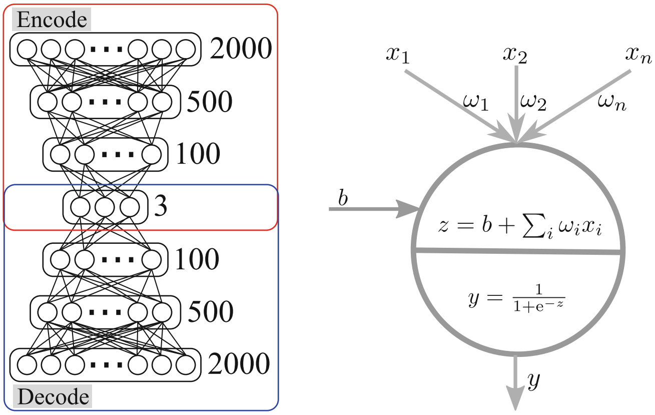

Fig. 2. Architecture of the supervised autoencoders used in this work, and detail of a single unit in it.

For training the autoencoders, we relied on standard gradient descent and back-propagation algorithms [7], which iteratively minimize the reconstruction square error e between m input training representations and their corresponding reconstructed outputs. This is,

$$e = \frac{1}{2} \cdot \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{n} \left(I_i^j - O_i^j \right)^2 , \tag{1}$$

where j is the index of the training example, and the dimensionality of the input I^j and its respective output O^j are indexed from $i = 1, \ldots, n$.

We trained all autoencoders during 1000 epochs, and used an auxiliary criterion for early stopping. Namely, training finished when one of the three following conditions was met:

- The maximum number of epochs was reached.
- The average sum of the square reconstruction error e was less than or equal to 10^{-3} in the last epoch.
- The average error e_t, of the t-th epoch, is very close to the average historic error h_t, which we computed using exponential smoothing as,

  $$h_t = \alpha h_{t-1} + (1 - \alpha)\, e_t, \tag{2}$$

  where, α is a parameter to control the amount of history that is tracked (we set $\alpha = 0.15$ in this work). Thus, we stopped training if $\varDelta\,(e) \leq 10^{-6}$, where,

  $$\varDelta\,(e) = |h_{t-1} - e_t|, \tag{3}$$

  where, $|\cdot|$ denotes absolute difference.

Although training itself is conducted without supervision, we refer to these autoencoders as "supervised" since the set of instances for training each autoencoder is defined under supervision, i.e., one autoencoder per class.

3.3 Dimensionality Reduction

Once the autoencoders are trained, shape representations are reduced by:

1. *Filter:* This step consists of passing a shape representation through both the encoder and the decoder to generate a reconstructed representation, which is expected to be a cleaned version of itself. In our case we consider one autoencoder per class, and we assume unawareness of the class of the test instance. Therefore, we pass its representation I through all autoencoders and generate a set $\{O^f\}$ of F different *filtered* outputs, as

   $$O^f = g^f\,(I)\,, \tag{4}$$

   where, $g^f\,(\cdot)$ denotes the full sequence of encoding and decoding performed by the f-th autoencoder.
2. *Max-pooling:* This consists of choosing the best candidate among the outputs generated by the set of autoencoders to be the final cleaned version. More precisely, we choose the output with the lowest reconstruction error with respect to the input representation. This is,

   $$O^* = arg \min_{O^f} \|I - O^f\|_2^2. \tag{5}$$

Finally, we choose the autoencoder that attains the lowest reconstruction error for a given input, and then use only its encoding phase to produce a short representation s. Mathematically,

$$s = \eth^*\,(I)\,, \tag{6}$$

where $\eth^*$ denotes the encoding phase of the autoencoder that produces O^*, which is chosen by Eq. (5).

Note that such short representations are suitable for both indexing and retrieval, as we will show in Sect. 5.

4 Hieroglyphic Shapes

The dataset used in this work is a subset of a larger collection currently under compilation by joint efforts of epigraphers and computer scientists. It comprises manually segmented and annotated glyph-blocks, among several more sign compounds of different granularity, which have been extracted from the extant Maya codices (folded bark-paper books) produced by the ancient Maya civilization within the Yucatan peninsula during the postclassic period (ca. 1100–1520 C.E). Although several thousand Maya hieroglyphic texts recorded on different media have been documented by explorers, archaeologists and researchers, the paramount importance of the codices lies in part in their extreme rarity, as the majority were destroyed by Spanish clerical authorities during colonial times, and only three of undisputed authenticity are preserved today at libraries at Dresden, Paris, and Madrid. The database is planned to be accessible to scholars as part of a future project publication.

Each record in the dataset consists of a single annotated glyph-block, which is a compound of several individual glyph-signs (glyph-blocks are often composed by one to six glyphs signs), which numbers and arrangement possibilities within the block can take a myriad different configurations that we are systematically investigating for Digital Palaeography and Sign-Encoding purposes. In turn, each individual sign is indicated by a unique code. The most commonly used catalog of glyph-signs is the Thomson catalog [20], where each glyph-sign is referred to by a consecutive number preceded by the letter ‘T’, e.g., T0024, T0106, etc.

Given that glyph-blocks are conformed by individual glyph-signs, we annotated them by the sequence of their constituting “members”. For instance, T0759b-T0025-T0181 and T0024-T1047a define two different classes, the former with 3 glyph-signs, and the later with 2. Note that, although the order of signs suggests by itself a sequential visual placement of the individual signs [8], there is not certitude of their actual location, and of whether they have been subject to scale or affine transformations. Nevertheless, our methodology is able to decode such visual variations, and produce accurate retrieval results. Furthermore, this definition of class poses two potential scenarios for partial matching: two classes with same individual glyph-signs in different order; and one class being a subclass of another one. However, we did not investigate partial matching cases in this work.

To produce the data used in this work, epigraphers in our team manually cropped glyph-block from the three Maya codices. For this work, a subset of glyph-blocks was chosen so that the percentage of images with visual noise was kept as in the complete dataset. Overall we defined two datasets: training and testing, both of them with the same 12 classes.

Regarding the training set, it corresponds to 102 instances manually cropped and cleaned by epigraphers. The test set, whose instances were only cropped but not cleaned, is formed by 780 glyph-blocks. Figure 3 shows the same glyph-block in both the training and test dataset, i.e., with and without the manual cleaning procedure.

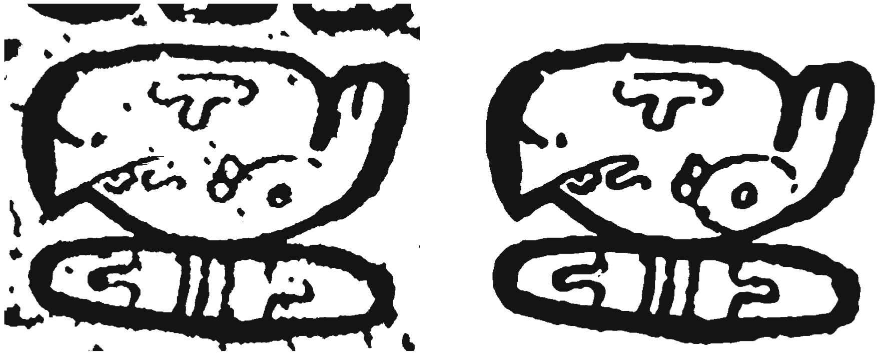

Fig. 3. Glyph-block with two glyph-signs: T0668 top and T0102 bottom. The same block is in both the training and test dataset, respectively, i.e., without and with the manual cleaning procedure.

Table 2. Total number of instances (glyph-blocks) in each dataset. Also, the minimum and maximum number of instances per class, and the respective average.

Dataset	Num. classes	Num. instances	Minimum	Average	Maximum
Train	12	102	5	$8.5 \pm \tilde{3}.6$	16
Test	12	780	8	65 ± 44.6	144

As shown in Table 2, the datasets are not totally balanced. However, the amount of instances per class remains within the same order, i.e., minimum and maximum within the same dataset. As small as these datasets might seem, epigraphers conducted a largely time consuming process to produce them. In fact, one long term objective of this work is to ameliorate such process.

5 Experimental Results

To assess the impact that the trained supervised autoencoders have on the shape representations, we computed the average intra-class variation that the three different shape representations produce on the training set, both before and after using the autoencoders. Namely, we computed this average intra-class variation as the average of the pairwise distance (Euclidean) between all elements within each class, and then averaged them across classes.

As shown in Table 3, the use of autoencoders helps producing representations with higher similarity within a visual class. In particular, the intra-class variation of the BoW representations does not change largely, while the neural representations conv5 and fc7 produce much lower intra-class variations after using the autoencoders.

We compared the retrieval performance obtained by:

- Raw: these are the three shape representations: BoW, conv5, and fc7.

Table 3. Average intra-class variation in the training dataset before and after cleaning the shape representations with the supervised autoencoders.

Method	Before (raw)	After (clean)
BoW	9.11×10^{-2}	3.03×10^{-3}
conv5	3.87×10^{3}	1.32×10^{1}
fc7	1.21×10^{2}	5.22×10^{-5}

- PCA: this consists in applying PCA to the input shape representations. For visualization purposes we chose the 3 principal components as output representations.
- AE3D: it results from applying dimensionality reduction by using only the encoding phase of the autoencoder, i.e., using Eq. (6). Thus a 3-dimensional vector.
- t-SNE: this is a dimensionality reduction method based on minimizing the Kullback-Leibler divergence between the distributions of representations in their original and reduced space [15]. We chose 3 components to make it comparable with PCA and AE3D.

Note that the dimensionality of Raw varies depending on the representation method, while PCA, AE3D, and t-SNE are the shortest representations with only 3 dimensions each.

We report our results as training and testing. For training, we used all elements in the training set as queries, one at a time, and compared them against all remaining instances also in the training set. This is done by using the L2 distance between pairs of shape representations. We proceed likewise for the elements in the test set, comparing them against all elements in the test set only. However, both the estimation of the visual vocabulary and the training of autoencoders were conducted using only the training set. We report the mean of the average precision computed using the 10 most similar glyph-blocks as retrieved by each of the methods (mAP@10).

Table 4 show the retrieval performance of the three shape representations before applying the dimensionality reduction techniques. As seen in Table 4, the neural-based representations work well for shape images, even when their parameters were learned using gray-scale images from the Imagenet. In particular, in the case of noisy data (the test set), off-the-shelf CNN representations from the conv5 layer outperform the other representations by a large margin (≈26−49 %).

As mentioned before, one of our goals is that of generating short representations that facilitate the visualization of the glyph-blocks for epigraphic analysis. Table 5 shows the mAP@10 results obtained after using the dimensionality reduction techniques listed at the beginning of this section. We used the training set here to learn the parameters of the autoencoder (AE3D). However, these results correspond to the test set only.

Table 5 shows that the use of t-SNE, with only 3 dimensions, improves the retrieval performance of the BoW approach, i.e., from 0.412 to 0.567; and that it produces retrieval results that are only slightly below for the neural-based

Table 4. Mean Average Precision before dimensionality reduction, i.e., using the 3 raw representations. These results were computed using the 10 most similar glyph-blocks as retrieved by each three shape representations (mAP@10).

	Training	Test
BoW	0.757	0.412
conv5	0.895	0.904
fc7	0.805	0.672

Table 5. Mean Average Precision (mAP@10) using dimensionality reduction techniques on the test set. Best result for each representation in bold.

	PCA	AE3D	t-SNE
BoW	0.390	0.515	**0.567**
conv5	0.569	0.564	**0.898**
fc7	0.346	0.256	**0.601**

representations, i.e., from 0.904 to 0.898 and from 0.672 to 0.601 respectively. Note that in general, t-SNE with 3 dimensions achieves higher retrieval performance than the other 3-dimensional approaches, i.e., PCA and AE3D. Namely, the t-SNE representations also have smooth transitions, e.g., erosion, among samples of a given class as reported in [2].

Regarding the performance attained with AE3D, one can see that this is an adequate approach to deal with bags of local descriptors. However, it results rather harmful for the case of neural representations. This behavior remains to be confirmed in a neural architecture that could include such compression layer, such that its training could happen during the classification-based training of the whole network, and not separately as we did here. In [4], the dimensionality of the last layer was decreased from 4096 to 128 with a small decrease in the performance (about 2 %) for the VGG-m net. As a future study, these encouraging results can motivate to add a dimensionality reduction layer at the end of the network structure and learn its parameters together with other parameters.

Also, an evaluation conducted using shallow autoencoders, of only one hidden layer of 3 units, resulted in very low performance, i.e., only 0.16 for the conv5 representation. Likewise, an attempt to use a single autoencoder for all classes produced very poor performance. This is due to fact that 3 dimensions are not enough for encoding enough information in a single model, which is the motivation for evaluating the performance of supervised autoencoders.

Figure 4 shows the average retrieval precision as a function of the standard recall for the three raw shape representations, and for their respective short representations obtained using t-SNE. These curves correspond to the generalization case, i.e., when the models are learned on the training dataset and then applied to the test data. Note that in general they are consistent with the results shown in Tables 4 and 5.

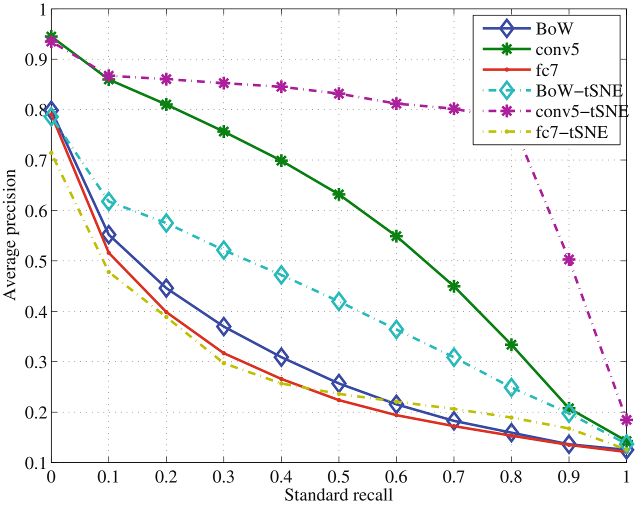

Fig. 4. Average retrieval precision of each method as a function of the standard recall.

The most relevant results from this experimentation are that: (1) complex shapes can be effectively indexed by neural representations (i.e., intermediate outputs of a convolutional neural network), even if they are trained on different datasets; and that (2) their dimensionality can be reduced up to 3 dimensions without too much harm to the retrieval performance, thus allowing for effective visualization of the complex shapes.

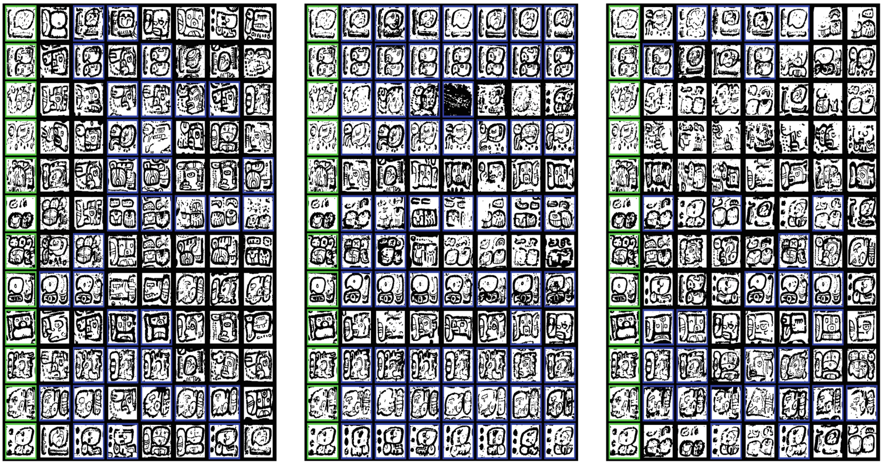

Fig. 5. Examples of retrieval results obtained on the Maya glyph-blocks with the three shape representations and the dimensionality reduction technique. From left to right: BoW-tSNE, conv5-tSNE, and fc7-tSNE. First column shows queries. Then from left to right are the most similar elements in descending order. The blue frame denotes a glyph-bock relevant to the query.

Finally, Fig. 5 shows examples of retrieval results obtained using the three shape representations and t-SNE. As seen in Fig. 5, more relevant glyph-blocks are retrieved by conv5-tSNE in comparison with the other two representations.

6 Conclusions

We proposed the use of neural representations of complex shapes, and the use of dimensionality reduction techniques for indexing Maya hieroglyphs, this with purposes of retrieval and visualization. Namely, we compared the retrieval performance obtained using the outputs of intermediate layers from a convolutional neural network, trained on the Imagenet dataset, and bag representations constructed from handcrafted robust local shape descriptors.

Our results show that this methodology is suitable to produce improved shape representations of very low dimensionality, i.e., up to 3 dimensions. In particular, the use of autoencoders is able to improve bag representations built upon handcrafted descriptors, although it does not have positive impact on the neural representations. Also, both bag and neural representations can be *compressed* to 3 dimensions with only a negligible drop in retrieval performance.

Two aspects of this work stand out. First, the successful use of neural-based representations learned on different datasets, which was important as the dataset of interest in this work is relatively small, thus resulting on the over-parametrization of the networks with respect to the dataset. Second, different from classical learning approaches, where evaluation is performed on datasets of considerably smaller size with respect to the training sets, we were able to achieve good performance with handcrafted features using a training set of about half the size of the evaluation set.

Finally, our methodology can be used for proposing known instances of Mayan glyphs, as candidates for deciphering new examples where visual noise hampers the decision of epigraphers.

Acknowledgments. This work was supported by the Swiss-NSF MAAYA project (SNSF–144238).

References

1. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
2. Can, G., Odobez, J.M., Gayol, C.P., Gatica-Perez, D.: Ancient Maya writings as high-dimensional data: a visualization approach. In: Digital Humanities (DH) (2016)
3. Can, G., Odobez, J.M., Gatica-Perez, D.: Evaluating shape representations for Maya glyph classification. ACM J. Comput. Cult. Heritage (JOCCH) (2016, accepted for publication)
4. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: delving deep into convolutional nets. In: British Machine Vision Conference (2014)

5. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Trans. Graph. **31**(4), 44:1–44:44 (2012)
6. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. Int. J. Comput. Vis. **59**(2), 167–181 (2004)
7. Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
8. Hu, R., Can, G., Gayol, C.P., Krempel, G., Spotak, J., Vail, G., Marchand-Maillet, S., Odobez, J.M., Gatica-Perez, D.: Multimedia analysis and access of ancient Maya epigraphy: tools to support scholars on maya hieroglyphics. IEEE Sig. Process. **32**(4), 75–84 (2015)
9. Hu, R., Pallan-Gayol, C., Krempel, G., Odobez, J.M., Gatica-Perez, D.: Automatic Maya hieroglyph retrieval using shape and context information. In: Proceedings of the ACM International Conference on Multimedia (ACM-MM) (2014)
10. Jolliffe, I.: Principal Component Analysis. Springer, New York (1986)
11. Krizhevsky, A., Hinton, G.: Using very deep autoencoders for content-based image retrieval. In: Proceedings of The European Symposium on Artificial Neural Networks (ESANN) (2011)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS) (2012)
13. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theor. **28**(2), 129–137 (2006)
14. Lu, S., Chen, Z., Xu, B.: Learning new semi-supervised deep auto-encoder features for statistical machine translation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (2014)
15. van der Maaten, L., Hinton, G.: Visualizing high-dimensional data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)
16. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
17. Roman-Rangel, E., Pallan, C., Odobez, J.M., Gatica-Perez, D.: Analyzing ancient Maya glyph collections with contextual shape descriptors. Int. J. Comput. Vis. **94**(1), 101–117 (2011)
18. Roman-Rangel, E., Wang, C., Marchand-Maillet, S.: SimMap: similarity maps for scale invariant local shape descriptors. Neurocomputing **175**(B), 888–898 (2016)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
20. Thompson, J.E.S.: A Catalog of Maya Hieroglyphs. University of Oklahoma Press, Norman (1962)
21. Wang, F., Kang, L., Li, Y.: Sketch-based 3D shape retrieval using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
22. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: a deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
23. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems (NIPS) (2014)
24. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part I. LNCS, vol. 8689, pp. 818–833. Springer, Heidelberg (2014)