Shape Representations for Maya Codical Glyphs: Knowledge-driven or Deep?

Gülcan Can
Idiap Research Institute, EPFL
Switzerland
gcan@idiap.ch

Jean-Marc Odobez
Idiap Research Institute, EPFL
Switzerland
odobez@idiap.ch

Daniel Gatica-Perez
Idiap Research Institute, EPFL
Switzerland
gatica@idiap.ch

ABSTRACT

This paper investigates two-types of shape representations for individual Maya codical glyphs: traditional bag-of-words built on knowledge-driven local shape descriptors (HOOSC), and Convolutional Neural Networks (CNN) based representations, learned from data. For CNN representations, first, we evaluate the activations of typical CNNs that are pretrained on large-scale image datasets; second, we train a CNN from scratch with all the available individual segments. One of the main challenges while training CNNs is the limited amount of available data (and handling data imbalance issue). Here, we attempt to solve this imbalance issue by introducing class-weights into the loss computation during training. Another possibility is oversampling the minority class samples during batch selection. We show that deep representations outperform the other, but CNN training requires special care for small-scale unbalanced data, that is usually the case in the cultural heritage domain.

CCS CONCEPTS

• **Applied computing** → **Arts and humanities**; • **Computing methodologies** → *Object recognition*; *Neural networks*;

KEYWORDS

Maya glyphs, convolutional neural networks, shape recognition

ACM Reference format:
Gülcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. 2017. Shape Representations for Maya Codical Glyphs: Knowledge-driven or Deep?. In *Proceedings of CBMI '17, Florence, Italy, June 19-21, 2017,* 6 pages.
https://doi.org/10.1145/3095713.3095746

1 INTRODUCTION

In this paper, we focus on learning shape representations for supervised classification of individual Maya codical hieroglyphs (in short, glyphs). Ancient Maya writing is composed of complex visual elements. One of the challenges for the visual recognition of the Maya writing is to decompose the script into its sub-elements, i.e. from a codex page to the glyph-blocks, and then to the individual glyphs. In our case, the experts provided the segmented glyph-blocks from the three Maya codices that survived up to date, namely Dresden [2], Madrid [3], and Paris codices [1]. In a previous study, we obtained individual glyphs from these segmented glyph-blocks by the help of the crowd [6]. Our crowdsourcing approach yielded a challenging medium-scale Maya codical sign dataset. In this paper, we analyze the visual content of these individually-segmented signs that are composed of several sub-parts with subtle visual characteristics.

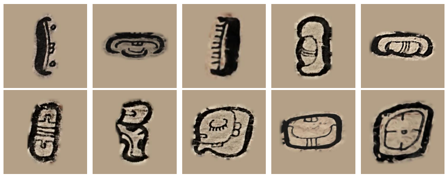

Figure 1: Segmented glyph samples from the 10-class experiment.

For the analysis of visual multimedia content, one of the common approaches is the standard bag-of-words built from knowledge-driven shape descriptors. Recently, representations learned from data directly by Convolutional Neural Networks (CNN) replaced this approach and were shown to be successful for various tasks [16], e.g. object classification, character recognition, image captioning.

This motivates us to investigate such CNN-based representation to handle cultural heritage data. One general issue in this domain is small to medium-scale amount of available data. We pose two important questions: 1) does the data source play a role (in terms of scale and nature of data), when leveraging pretrained CNN activations for the target data? 2) can we train a specialized CNN from scratch with unbalanced medium-scale data in cultural heritage domain? For the data source, we investigated natural images of the objects and sketches of everyday objects. In this sense, we utilized the VGG-16 net pretrained on ImageNet (1M RGB images for 1K objects in the WordNet ontology) [9], and Sketch-a-Net pretrained on hand-drawn sketches of 250 everyday objects (20K binary images) [11]. These datasets are balanced per class. As one important point is the scale of data for learning representations with CNNs, we populate the available individual glyph images with data augmentation such as translation and scaling transformations. With the populated data, we train a CNN that is inspired from a model for sketch classification, i.e. sketch-a-net [29]. On the other hand, we considered assigning class-weights while computing the loss during training for favoring the minority classes. We compare our results with a traditional bag-of-words pipeline [22], and with the features extracted from the pretrained networks, as they are shown to be powerful baselines for many computer vision tasks [23].

The contributions presented in this paper are as follows:

CBMI '17, June 19-21, 2017, Florence, Italy

ACM ISBN 978-1-4503-5333-5/17/06…$15.00
https://doi.org/10.1145/3095713.3095746

(1) evaluating the standard bag-of-words approach with local HOOSC shape descriptor for the crowdsourced individual codical glyphs,
(2) evaluating the features extracted from the pretrained models for all the valid crowdsourced individual glyphs,
(3) training a CNN from scratch for Maya glyph classification,
(4) attempting to handle data imbalance issue during training.

The rest of the paper is organized in five sections. Section 2 presents the related work about CNNs for multimedia applications focusing on the cultural heritage analysis. Section 3 explains the Maya writing system, and describes the crowdsourced individual glyph dataset exploited in this paper. Section 4 introduces the methodology. Section 5 discusses the results, and section 6 concludes the paper.

2 RELATED WORK

In this sections, we present three lines of related research: knowledge-driven shape representations, data-driven representations learned by CNNs, and sketch-specific representations.

Knowledge-driven representations. Traditional shape descriptors, i.e. scale-invariant feature transform (SIFT) [18], Histogram of Oriented Gradients (HOG) [8], and shape context (SC) [4], are commonly used for visual object recognition tasks in a standard bag-of-words (BoW) pipeline [26]. As an application of cultural heritage data analysis, Roman-Rangel et. al. obtained promising retrieval results for Maya monumental glyphs with the bag-of-words representation of the Histogram of Orientation Shape Context (HOOSC) descriptor [22]. The proposed HOOSC descriptor is a combination of HOG and SC descriptors. Similarly, Hu et. al. applied the 2-ring HOOSC descriptor [21] for both monumental and codical Maya glyph retrieval [13]. Even though the glyphs carved on the monuments and the glyphs brushed on the folded codices have visual differences, HOOSC-BoW representation has been shown as a good baseline for both types of the data.

Furthermore, Franken et. al. showed that HOOSC descriptor yields competitive and promising results for Egyptian glyph recognition in a comparative analysis with other traditional descriptors, i.e. variants of the HOG and SC [12].

Data-driven deep representations. On the other hand, Convolutional Neural Networks (CNN) demonstrated strong results recently in visual recognition tasks, such as object classification on ImageNet dataset [15, 24]. Motivated by the common visual structures learned by CNNs in the first layers, several transfer learning approaches reutilized and analyzed the effectiveness of the pretrained CNN representations on different datasets [10, 23, 27]. Razavian et. al. pointed out that the penultimate activations of a CNN, specifically AlexNet [15], is a strong baseline for visual recognition tasks.

In this respect, the activations of a VGG-M network [7] pretrained on ImageNet dataset were explored to index Maya codical glyph-blocks (composed of several individual glyphs) [20]. Authors concluded that the middle-layer activations of VGG-M followed by a nonlinear dimensionality reduction method were powerful representations for the retrieval of glyph-blocks, outperforming HOOSC-BoW representations in this scenario. Since feature extraction from pretrained CNN models is a standard baseline approach nowadays, we investigate such pretrained model activations in our individual glyph classification task against the traditional HOOSC-BoW representation.

Sketch-specific representations. As considerable amount of the Maya signs look like everyday-objects or body parts, shape representations learned for sketch recognition tasks are also highly related. Eitz et. al. introduced a relatively large-scale sketch dataset of 250 everyday-objects [11]. This dataset enabled extensive analysis of shape representations including the traditional ones [11] as well as neural representations learned from these sketches [5, 28, 29].

Yu et. al. proposed a multi-scale ensemble CNN model called Sketch-a-Net, that derives from the AlexNet model with several modifications for handling the sparse shapes, e.g. larger filter sizes in the first layers [29]. The Sketch-a-Net model is reported to outperform human annotators for the sketch label prediction task. In a later study, Yu et. al. showed that training a single-scale competitive CNN model is feasible with data augmentation such as local and global deformations of the contours as well as standard translation and scaling transformations [28]. This motivated us to train a similar CNN model from scratch for Maya individual glyphs by the help of data augmentation.

3 DATA

3.1 Maya Writing System

The Maya writing system is quite visual, and composed of complex logograms and syllabograms as opposed to stroke-based or continuous scripts of other languages, i.e. Chinese or Arabic scripts. In a recent catalog, Macri and Looper categorizes Maya glyphs into semantic groups such as animals, body parts, and faces. Other categories are not straightforward to interpret as everyday-objects, however they are also classified with some visual hints like square contour, with or without inner symmetry, elongated shapes, or variable number of components.

As can be seen in everyday objects, the samples from each Maya category may exhibit high within-class variance and low between-class differences. Due to the era, place, and artistic changes, Maya glyphs from one category may look relatively different except some specific “diagnostic” parts. Similarly, between two similar-looking classes, the difference may be quite subtle in the diagnostic local parts such as eye or teeth in the head signs. Nevertheless, learning global patterns like shape contours (rectangular, head-shape, elongated) or local patterns (small circles, eyes, teeth) across classes, as CNNs are truly capable of achieving, will most probably benefit the recognition task.

3.2 Crowdsourced Maya Glyph Segments

In [6], a Maya Codical dataset is curated by the help of the crowd segmenting each glyph from glyph-blocks in three survived codices. The original data was provided by our project partners (see [13]). As these codices are from the post-classical era, within-class variance is relatively less than the monumental glyphs coming from different eras. However, it is possible to observe the stylistic differences. The generated dataset is quite challenging as the number of samples per class is low due to the lack of available data.

Furthermore, the visual differences can be quite subtle, such as just the orientation of signs. Fig. 1 illustrates samples from the ten

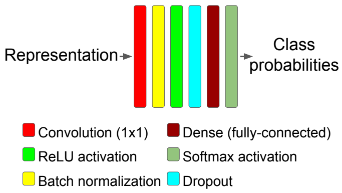

Figure 2: The shallow CNN model for classification of the representations obtained with the method (a, BoW) and (b, pretrained CNN).

classes with highest number of samples per class. For example, the only visual difference between the last two glyphs in top row is orientation. These examples demonstrate that the classification task is not trivial even in 10-class case with medium amount of data.

4 METHODOLOGY

To assess the shape representations for glyph recognition tasks, we evaluated (a) the traditional bag-of-words representation of a knowledge-driven local shape descriptor (HOOSC), (b) the knowledge transfer approach from a pretrained network, (c) learning the representation by training a convolutional neural network from scratch. We describe each of these methods below.

4.1 Bag-of-Words Approach

We followed the same pipeline as proposed for the retrieval task in [13] with an additional normalization factor at the end. The steps are as follows.

HOOSC Descriptor Extraction. After binarizing the glyph segments via global Otsu's method [19] (threshold is determined on the corresponding glyph-block image), and applying morphological operations (i.e. closing), we obtain the glyph skeletons. Skeletons are used to select pivot points, and compute the HOOSC descriptor around each pivot point. To define the local neighborhood while computing the HOOSC descriptor, we used 2-rings and the whole glyph context. Hence, the HOOSC descriptor around a pivot point counts the normalized frequencies of the skeleton points in two radial circles (8 orientations), and quantize them in 8 bins. This process produces a 128-dimensional local descriptor around each pivot point. We did not consider concatenating relative spatial location of the pivots here. We, randomly, selected 400 or more ($0.1 * N_{skeletonpoints}$) pivots from each glyph skeleton if possible, otherwise we used all the skeleton points as pivots.

Building the Dictionary. After extracting the local HOOSC descriptors for each glyph, we sampled 80% of the glyphs randomly. From this set of glyphs, we sampled 10% of the HOOSC descriptors of each glyph to build the dictionary by applying k-means with 4000 cluster centers.

Assigning Descriptors to the Codebook. After computing the dictionary with vocabulary size 4000, we assign each HOOSC descriptor of each glyph to their closest cluster center (or word in the dictionary) with $L1$ distance. Therefore, for each glyph, we obtain a codebook that corresponds to the frequencies of closest words of its HOOSC descriptors in the dictionary. Final representation is addressed as HOOSC-bag-of-words (HOOSC-BoW), and has 4000 dimensions.

Normalization. Due to the nature of the BoW computation, i.e. hard-assignment, the HOOSC-BoW representation is distributed among the 4000 dimensions with a constraint on the dimensions summing up to 1. A normalization of this representation with a scaling factor is needed to obtain a reasonable comparison with CNN activations. Therefore, we, first, normalized the BoW vectors of each glyph with the corresponding max value, i.e. making the max value of each vector 1, instead of sum of the vector being 1, and then scaled the BoW vectors with a constant scalar to match the maximum activation value of the pretrained CNN features.

Classification. The BoW features are used as input to a shallow neural network as illustrated in Fig. 2. This network with two fully-connected (FC) layers has 1024 filters in its first FC layer. We applied ReLU activation between two FC layers as well as batch normalization, and dropout method with 0.5 rate. The final class probabilities are determined by the softmax activation at the end.

4.2 Pretrained CNN Features

The CNNs pretrained on large-scale datasets, i.e. ImageNet, are used as feature extractors by feedforwarding the image of interest, and gathering the activations at different layers of the network [10, 23, 27]. Razavian et. al. and Donahue et. al. reported the penultimate activations before softmax classifier as the strong baselines for transferring knowledge in several vision tasks. Furthermore, Yosinski et. al. showed that the middle-layer activations are more generic than the last-layer ones, and may be more applicable to the data with different nature (e.g. man-made vs. natural objects).

With this motivation, we forward the glyph segments in our dataset through a pretrained network, and collect the activations at the end of the last convolutional block. We consider these activations as our pretrained CNN features.

Considered Networks. We considered the VGG-16 network [25] pretrained on ImageNet dataset, and the Sketch-a-Net [29] pretrained on 250-class binary sketch images [11].

VGG-16 is a deep CNN model, i.e. 16 layers, that takes the initiative from the LeNet structure (3-convolutional layers) [17]. VGG-16 is shown to be competitive on the ImageNet dataset before the inception module and residual connections were introduced. We passed our RGB glyph images from the pretrained VGG-16, and extracted the activations from the last (5^{th}) convolutional layer.

Sketch-a-Net is adapted from the AlexNet model [15] for handling sparse sketch images. We retrained the single-scale single-channel version of the Sketch-a-Net model with an important modification: adding a batch normalization (BN) layer [14] after each convolutional and dense layer. Batch normalization is an effective way to prevent the optimizer oscillating with the gradient updates in different orientations, since it normalizes the data to have the same distribution. This approach speeds up the training and improves the results. The modified Sketch-a-Net obtained competitive results on a random split of the sketch dataset (72.2% accuracy). We used this model to extract the activations of the binarized version of our glyph images. This model is also used in method (c) below, and is illustrated in Fig. 3. As we consider the hand-drawn sketch data is close in nature to our glyph images that are brushed on the codices, we extracted the penultimate activations (at the end of block 7 in Fig. 3) in this case. For assessing these representations, the same network as in Section 4.1 is used to do classification (Fig. 2).

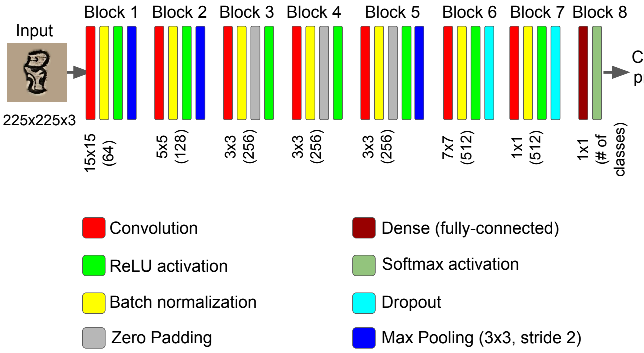

Figure 3: The modified Sketch-a-Net used in the method (c).

Table 1: The number of available original glyph segments for the classification tasks.

		# of classes			
		10	**50**	**100**	**150**
# of samples	**min**	210	49	17	5
	mean	257.1	133.5	81.76	58.2
	median	244	103.5	50	27

4.3 CNN Training

As our main contribution, we trained a Sketch-a-Net model with additional batch normalization layers with individual Maya codical glyphs from scratch. Fig. 3 illustrates the model structure that is composed of eight blocks. The first block starts with a 15x15 convolution (red) with a stride of 3 pixels and 64 filters (activation maps). This is the main change of the Sketch-a-Net model from the AlexNet structure for handling sparse shapes. Another change is the number of filters at each convolution layer. As in the original paper, we applied 0.5 dropout rate at both indicated layers (yellow).

Considering the amount of our data, we augmented our images with random transformations on-the-fly during training. This differs from the previous method (b), in which we only provide the available data to the pretrained network without any transformation. Data augmentation with random transformations is one of the ways to deal with the lack of data while training CNNs.

Applying Class Weights. On the other hand, the data imbalance issue may be handled by introducing class weights into the loss computation during training. We computed the class weights as a proportion of the maximum number of samples per class to the number of samples per each class. In this way, if the samples from the minority classes are misclassified, the loss function would increase more dramatically. This way, the optimizer would favor the minority classes, as they are picked, randomly, less often compared to the majority classes during the batch-based training.

Oversampling. We also experimented with oversampling by synthesizing the data with random transformations to provide equal number of samples per class during training. This approach would make the samples from the under-represented classes more visible to the optimizer, i.e. to have the same chance to be picked, as samples in a batch are picked randomly for a gradient update computation.

5 SETTINGS AND RESULTS

5.1 Experimental Settings

5.1.1 Data. We use the individual valid glyph segments curated via crowdsourcing in [6]. We use the subset for the selected 150 classes with the most number of samples. The authors point out data imbalance as one of the challenges of the dataset. Accordingly, they prepared several classification cases (easy to difficult) with different number of classes and number of samples per class. In each case, they use the same number of samples per each class, and provide five folds of the data containing randomly-picked samples from the available glyphs. Here, we are using all the available samples in the selected 150 classes instead. The folds are divided randomly into training validation, and test sets (roughly 60%-20%-20%). Note that these sets of the glyph segments are populated with four background colors (3 static, 1 dynamic RGB color).

5.1.2 Tasks and performance measures. We focus on four tasks (a subset of those in [6]): 10-, 50-, 100-, and 150-class glyph classification. The basic statistics about the original samples per class is provided in Table 1 below. Note that the 10, 50, and 100 classes are chosen from 150-class set such that they have the most number of samples per class. Therefore, the maximum number of original glyphs per class in all the classification tasks is 381.

We report the average sample-based test accuracy across 5-folds, along with the top-five accuracy (noted as micro-average). We also report the average class-based accuracies (noted as macro-average). The difference in the micro- and macro-averages indicate overfitting, and data imbalance problems during training.

5.1.3 Data preparation and augmentation. During training of the shallow network over the pretrained features, we only rely on the available samples. However, during training the modified Sketch-a-Net from scratch, we apply on-the-fly random geometric data augmentation, comprising rotation (within $[-15, 15]$ degrees), vertical and horizontal translation ($+/-0.1\times$ image width), and zooming (scale within $[0.8, 1.2]$). Similar to [25], the image width is set to 224 pixels when using the VGG-16 net; whereas for the Sketch-a-Net case it is set to 225 pixels as in the original paper [29].

5.1.4 Training. Adam optimizer with the learning rate 10^{-4} is utilized during training, while following an early-stopping approach with the patience factor of 20 epochs, i.e. terminating training if the loss does not decrease for 20 epochs. We applied model check-pointing (keeping track of the parameters that result in highest validation accuracy during the optimization) during training, as the maximum number of epochs is set to 1000. However, we observed that the training and the validation accuracies reach $85-95\%$, in average, in less than 100 epochs for almost all the cases due to the early-stopping approach. Note that the number of epochs is determined empirically, and higher number of epochs may result in higher accuracies than those reported here.

5.2 Classification Results

Our experiments involve four different methods. More specifically, these methods are as follows: (a) Normalized HOOSC-BoW; (b.1) Sketch-a-Net pretrained on binary sketch images (Sketch-a-Net-binary-B); (b.2) VGG-16 net pretrained on ImageNet (VGG-16-RGB-B); (c) Sketch-a-Net trained from scratch (Sketch-a-Net-S).

Table 2: Average sample-based top-1 (T-1), top-5 (T-5) accuracies, and average class-based accuracies (CA) for the test set.

	# of classes											
	10			**50**			**100**			**150**		
Method	**T-1**	**T-5**	**CA**	**T-1**	**T-5**	**CA**	**T-1**	**T-5**	**CA**	**T-1**	**T-5**	**CA**
(a) HOOSC-BoW	68.7	95.3	67.4	49.8	75.9	42.1	43.0	67.6	26.6	40.7	63.9	19.2
(b.1) Sketch-a-Net-binary-B	77.5	97.2	76.5	59.4	84.2	54.0	54.3	78.1	42.0	52.0	76.3	36.5
(b.2) VGG-16-RGB-B	87.8	98.6	**87.6**	83.3	94.5	**80.9**	79.0	92.2	**68.8**	76.5	89.7	**56.2**
(c.1) Sketch-a-Net-S	**91.7**	**99.4**	77.1	89.1	96.9	51.9	86.7	**95.9**	26.9	84.2	94.4	16.0
(c.2) Sketch-a-Net-S (class weights)	90.7	99.2	75.7	**89.7**	**97.0**	52.4	**87.2**	**95.9**	26.8	**84.8**	**94.5**	17.7

Table 2 presents the results we obtained with these methods. Compared to the undersampling approach applied in the previous study in [6], our sample-based accuracy (T-1) results of method (b.2) are consistently higher. This proves that utilizing all the data helps to improve performance, as the data amount increases considerably.

The results of method (c) shows an overfitting problem towards the classes with many samples, as the difference in the sample-based (T-1) and class-based (CA) accuracies increases with the number of classes. We, especially, spotted that some classes with few samples in 100- and 150-class cases are not predicted correctly at all, therefore the class averages dropped dramatically.

Interestingly, we observe that the VGG-16 activations, i.e. method (b.2), are still able to maintain the decrease in a slower pace than the method (c). Hence, we hypothesize that these activations might be more generic as they are learned from a larger-scale corpus (ImageNet) than our glyph corpus. When we compare results of the method (b.1) and (b.2), we observe that the pretrained features from a CNN trained with a large corpus are more promising, even though the sketch data looks more relevant to our glyph examples. As a note, this difference in performance may also originate from the difference in the pretrained CNN structures, as VGG-16 is deeper and have a larger model capacity compared to Sketch-a-Net.

Another interesting point is the class-based results of the methods (b.1) and (c.1) being similar for 10- and 50-class cases (where we do not observe zero-performing class). However, sample-based accuracy (T-1) of the method (c.1) is higher than the method (b.1) with a large margin.

The results of the method (c.2) demonstrate that applying class-weights into the loss function alone is not enough for the optimizer to learn classifying the minority class correctly. As the batch-preparation is still done randomly among all the samples, we hypothesize that the samples from the minority classes may not be seen by the optimizer as often as the majority classes to have a real impact on the loss function. This is why the results are only slightly different as the required epochs for both of the method (c.1) and (c.2) are similar, before the early-stopping mechanism terminates training and the optimizer gets exposed to more minority class samples. These results show that oversampling the minority classes is essential during training.

To oversample, we augmented some randomly-picked samples from each class until each has 1000 samples in the training set. With this operation, we observed around 3% improvement in the class-based average of the 10-class case of the method (c).

Bottom lines. From our experiments, we observed that 1) leveraging all the data during CNN training improves performance (compared to the previous study); 2) computing the class-based average along the sample-based performance brings insights to training process; 3) the activations from the pretrained CNNs generalize better for the classes with few samples; 4) oversampling helps the results for the unbalanced data case.

6 CONCLUSION

To sum up, this paper assessed the knowledge-, and data-driven shape representations for individual Maya codical glyph recognition. We analyzed the performance of the standard bag-of-words approach over the local HOOSC descriptors along with the CNN-based representations for the classification tasks with different level of difficulty, i.e. medium-scale data with few classes (10-class) to small-scale data with many classes (150-class). The CNN-based representations are either transferred from related pretrained networks, i.e. the VGG-16 net on the ImageNet data, or the Sketch-a-Net from the 250-class sketch dataset, or they are learned from the codical glyph data directly by training a modified Sketch-a-Net structure with batch normalization layers from scratch.

We observed that leveraging all the available valid segments yields better performance compared to the undersampling approach utilized in the previous study [6]. An important note is that the imbalance in the data and the flaws in the training can be noticed by checking the macro-average (class-based) performance along with the micro-average (sample-based) performance. The decrease in the average class performances with the CNN trained from scratch for the 100- and 150-class cases is due to some under-performing classes with few samples. We also showed that oversampling during training step by the help of the data augmentation improves the results. An interesting is that the pretrained VGG-16 net activations generalizes better than the specialized CNN for the classes with few samples, i.e. generates higher average class accuracies, even though the sample-based accuracies are vice-versa.

Overall, we concluded that the learned CNN representations outperform the standard knowledge-driven HOOSC-BoW representation. These representations may be good candidates for retrieval tasks as well, with over 94% top-5 sample-based accuracies.

For future work, exploring the data-balancing approaches for training from few samples with specialized loss functions, i.e. triplet loss, might be interesting. Another possible direction is to incorporate the recent advancements, i.e. inception modules and residual connections to the CNN training process.

Acknowledgments. This work was supported by the SNSF through the MAAYA project. The original images, that are adapted in Fig. 1 and 3, were created by Carlos Pallan Gayol for the MAAYA project.

REFERENCES

[2] 1880. Dresden Codex. http://digital.slub-dresden.de/werkansicht/dlf/2967/1/. (1880).

[3] 1883. Madrid Codex. http://www.famsi.org/mayawriting/codices/madrid.html. (1883).

[1] 1887. Paris Codex. http://gallica.bnf.fr/ark:/12148/btv1b8446947j. (1887).

[4] Serge Belongie, Jitendra Malik, and Jan Puzicha. 2000. Shape context: A new descriptor for shape matching and object recognition. In *Conference on Advances in Neural Information Processing Systems*, Vol. 2. 3.

[5] Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. 2016. Evaluating Shape Representations for Maya Glyph Classification. *ACM Journal on Computing and Cultural Heritage (JOCCH)* 9, 3 (September 2016).

[6] Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. 2017. *Maya Codical Glyph Segmentation: A Crowdsourcing Approach.* Research Report. Idiap.

[7] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *BMVC*.

[8] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition*, Vol. 1. IEEE, 886–893.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*. IEEE, 248–255.

[10] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition.

[11] Mathias Eitz, James Hays, and Marc Alexa. 2012. How Do Humans Sketch Objects? *ACM Trans. Graph.* 31, 4, Article 44 (jul 2012), 10 pages.

[12] Morris Franken and Jan C van Gemert. 2013. Automatic egyptian hieroglyph recognition by retrieving images as texts. In *International Conference on Multimedia*. ACM, 765–768.

[13] Rui Hu, Gulcan Can, Carlos Pallan Gayol, Guido Krempel, Jakub Spotak, Gabrielle Vail, Stephane Marchand-Maillet, Jean-Marc Odobez, and Daniel Gatica-Perez. 2015. Multimedia Analysis and Access of Ancient Maya Epigraphy. *Signal Processing Magazine* 32, 4 (jul 2015), 75–84.

[14] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. of International Conference on Machine Learning*. 448–456.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in NIPS*. 1097–1105.

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[17] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[18] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 2 (2004), 91–110.

[19] Nobuyuki Otsu. 1975. A threshold selection method from gray-level histograms. *Automatica* 11, 285-296 (1975), 23–27.

[20] Edgar Roman-Rangel, Gulcan Can, Stephane Marchand-Maillet, Rui Hu, Carlos Pallan Gayol, Guido Krempel, Jakub Spotak, Jean-Marc Odobez, and Daniel Gatica-Perez. 2016. Transferring Neural Representations for Low-dimensional Indexing of Maya Hieroglyphic Art. In *ECCV Workshop on Computer Vision for Art Analysis*.

[21] Edgar Roman-Rangel, Jean-Marc Odobez, and Daniel Gatica-Perez. 2013. Evaluating Shape Descriptors for Detection of Maya Hieroglyphs. In *Mexican Conference on Pattern Recognition*.

[22] Edgar Roman-Rangel, Carlos Pallan, Jean-Marc Odobez, and Daniel Gatica-Perez. 2011. Analyzing ancient maya glyph collections with contextual shape descriptors. *IJCV* 94, 1 (2011), 101–117.

[23] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *CVPR Workshops*.

[24] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[25] K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[26] Josef Sivic and Andrew Zisserman. 2003. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *ICCV (ICCV '03)*. 1470–. http://dl.acm.org/citation.cfm?id=946247.946751

[27] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in NIPS*. 3320–3328.

[28] Qian Yu, Yongxin Yang, Feng Liu, Yi-Zhe Song, Tao Xiang, and Timothy M. Hospedales. 2016. Sketch-a-Net: A Deep Neural Network that Beats Humans. *IJCV* (2016), 1–15.

[29] Qian Yu, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy Hospedales. 2015. Sketch-a-net that beats humans. In *Proc. of BMVC*. 7.1–7.12.