Visual Analysis of Maya Glyphs via Crowdsourcing and Deep Learning

THÈSE N^{O} 8058 (2017)

PRÉSENTÉE LE 27 NOVEMBRE 2017
À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE L'IDIAP
PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Gülcan CAN

acceptée sur proposition du jury:

Dr D. Gillet, président du jury
Prof. D. Gatica-Perez, Dr J.-M. Odobez, directeurs de thèse
Prof. A. del Bimbo, rapporteur
Prof. R. Ingold, rapporteur
Prof. J.-Ph. Thiran, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2017

Abstract

In this dissertation, we study visual analysis methods for complex ancient Maya writings. The unit sign of a Maya text is called *glyph*, and may have either semantic or syllabic significance. There are over 800 identified glyph categories, and over 1400 variations across these categories. To enable fast manipulation of data by scholars in Humanities, it is desirable to have automatic visual analysis tools such as glyph categorization, localization, and visualization.

Analysis and recognition of glyphs are challenging problems. The same patterns may be observed in different signs but with different compositions. The inter-class variance can thus be significantly low. On the opposite, the intra-class variance can be high, as the visual variants within the same semantic category may differ to a large extent except for some patterns specific to the category. Another related challenge of Maya writings is the lack of a large dataset to study the glyph patterns.

Consequently, we study *local shape representations*, both knowledge-driven and data-driven, over a set of frequent syllabic glyphs as well as other binary shapes, i.e. sketches. This comparative study indicates that a large data corpus and a deep network architecture are needed to learn data-driven representations that can capture the complex compositions of local patterns.

To build a large glyph dataset in a short period of time, we study a *crowdsourcing* approach as an alternative to time-consuming data preparation of experts. Specifically, we work on individual glyph segmentation out of glyph-blocks from the three remaining codices (i.e. folded bark pages painted with a brush). With gradual steps in our crowdsourcing approach, we observe that providing supervision and careful task design are key aspects for non-experts to generate high-quality annotations. This way, we obtain a large dataset (over 9000) of individual Maya glyphs.

We analyze this crowdsourced glyph dataset with both knowledge-driven and data-driven visual representations. First, we evaluate two competitive knowledge-driven representations, namely Histogram of Oriented Shape Context and Histogram of Oriented Gradients. Secondly, thanks to the large size of the crowdsourced dataset, we study *visual representation learning* with deep Convolutional Neural Networks. We adopt three data-driven approaches: assessing representations from pretrained networks, fine-tuning the last convolutional block of a pretrained network, and training a network from scratch.

Finally, we investigate different *glyph visualization* tasks based on the studied representations. First, we explore the visual structure of several glyph corpora by applying a non-linear dimensionality reduction method, namely t-distributed Stochastic Neighborhood Embedding, Secondly, we propose a way to inspect the discriminative parts of individual glyphs according to the trained deep networks. For this purpose, we use the Gradient-weighted Class Activation Mapping method and highlight the network activations as a heatmap visualization over an input image. We assess whether the highlighted parts correspond to distinguishing parts of glyphs in a perceptual crowdsourcing study.

Overall, this thesis presents a promising crowdsourcing approach, competitive data-driven visual representations, and interpretable visualization methods that can be applied to explore various other Digital Humanities datasets.

Keywords: Maya civilization, hieroglyphs, cultural heritage, crowdsourcing, shape classification, local shape representations, convolutional neural networks, visualization

Résumé

Cette thèse aborde des méthodes d'analyse visuelle d'écrits mayas anciens. L'écriture maya, à base de *hiéroglyphes*, est complexe d'un point de vue visuel. L'unité de base du texte maya est appelée un *glyphe*, et peut avoir une signification spécifique ou être liée à la prononciation (syllabique). Avec plus de 800 catégories et plus de 1400 variations à l'intérieur de ces catégories, il est appréciable pour les chercheurs en sciences sociales d'avoir à disposition des outils automatiques d'analyse visuelle, comme la classification, la localisation, ou encore la visualisation.

L'analyse et la reconnaissance de glyphes sont des problèmes difficiles : au sein des différentes catégories, certains motifs peuvent généralement être observés dans des compositions différentes (faible variance *inter*-classes). De plus, il est possible d'observer des variantes très différentes dans une même catégorie (variance *intra*-classe large), qui n'inclue cependant pas les motifs caractéristiques de la catégorie en question. Un autre défi posé par l'écriture maya est l'absence de base de données suffisamment grosse pour examiner les motifs des glyphes.

Basé sur ces observations, nous nous intéressons dans un premier temps aux *représentations des formes locales*, en appliquant des méthodes basées sur des connaissances et des méthodes exploitant les données. Cette analyse est conduite sur un ensemble de glyphes syllabiques fréquents, ainsi que sur d'autres formes binaires, en l'occurrence des sketches. Cette étude comparative indique que pour apprendre des représentations à partir de données capables de capturer les compositions complexes de motifs locaux, un large corpus de données *et* une architecture profonde de réseau sont essentiels.

Afin de construire une base de données de taille importante rapidement, nous proposons une approche à base de *crowdsourcing* (production participative) qui exploite les trois dernières codices dont l'existence est connue. Notre approche est une alternative viable à la procédure standard qui consiste à préparer les données par des experts et s'avère chronophage. Une succession d'étapes nous mène à la conclusion qu'un design minutieux de la tâche à accomplir ainsi qu'une supervision adéquate sont les pierres angulaires dans la construction d'annotations de haute qualité par des non experts. Il en résulte une large base de données de glyphes mayas individuels (plus de 9000).

Nous analysons ce corpus à l'aide de représentations visuelles basées sur les connaissances dans un premier temps, et basées sur les données ensuite. En premier lieu, nous évaluons deux méthodes compétitives basées sur les connaissances, à savoir l'Histogramme de Contexte de Formes Orientées (HOOSC), et l'Histogramme de Gradients Orientés (HOG). Dans un second temps, grâce à la taille du corpus, nous sommes en mesure d'étudier l'apprentissage de représentations visuelles grâce aux Réseaux de Neurones Convolutionnels profonds (CNNs). Nous proposons trois approches distinctes : l'évaluation de représentations de réseaux pré-entraînés, l'ajustement des paramètres du dernier bloc convolutionnel d'un réseau pré-entraîné, et l'entraînement complet d'un réseau.

Enfin, nous nous intéressons à la *visualisation de glyphes* basée sur les représentations évoquées précédemment. Dans un premier temps, pour comprendre la structure visuelle de plusieurs corpus de glyphes, nous appliquons une réduction dimensionnelle non-linéaire, nommée t-distributed Stochastic Neighborhood Embedding (t-SNE). Dans un second temps, nous proposons un moyen d'examiner les segments discriminatifs de glyphes individuels, selon les réseaux entraînés. Pour cela, nous utilisons le Mapping d'Activation de Classes pondéré par des Gradients (Grad-CAM), et mettons en évidence les activations du réseau en superposant leur visualisation (en utilisant une carte d'intensité) à une image donnée en entrée du système. Nous évaluons la correspondance entre ces segments sélectionnés et les segments distinctifs de glyphes à travers une étude perceptive de crowdsourcing.

De manière générale, cette thèse propose une approche de crowdsourcing, des représentations visuelles compétitives basée sur les données, et des méthodes de visualisation interprétables. Ces contributions obtiennent des résultats prometteurs qui permettent d'envisager leur application à d'autres corpus d'Humanités Numériques.

Mots clefs : Civilisation maya, hiéroglyphes, héritage culturel, crowdsourcing, classification de formes, représentations de formes locales, réseaux de neurones convolutionnels, visualisation

Acknowledgements

First, I would like to thank Daniel Gatica-Perez and Jean-Marc Odobez for their supervision and the opportunity they provided for me to pursue my doctoral studies in Idiap and EPFL. I sincerely appreciate their guidance, valuable feedback, and professional attitude that helped me grow as a researcher.

Secondly, I would like to acknowledge the Swiss National Science Fundation that had funded our work through the MAAYA project. Furthermore, I would like to thank all the project partners, collaborators, and colleagues that made it possible to accomplish the work presented in this thesis. Specifically, I am very grateful to Carlos Pallán Gayol, Guido Krempel, and Jacub Spotak for sharing their knowledge on the Maya sources, generating the glyph-block dataset, and providing the glyph annotations; Gabrielle Vail and Christine Hernández for their collaboration; Edgar Roman-Rangel, Stephane Marchand-Maillet, and Rui Hu (Idiap) for fruitful discussions.

I would like to thank to my thesis committee, Prof. Alberto del Bimbo, Prof. Rolf Ingold, Prof. Jean-Philippe Thiran and Dr. Denis Gilet for their constructive comments that helped me to polish my thesis.

I would like to thank the system team and the administration team in Idiap for their help on various matters all these years.

I appreciate all the fun times that I got to spend with many great people in Idiap and EPFL. Thanks to some of these people, I got to learn skiing, and got motivated to do long hikes in the weekend. That is how I started to enjoy the breathtaking sceneries around Martigny. All these outings, other social gatherings, and the fun trips around Europe would be as memorable and significant to me as my work in this thesis. I would like to thank you all for your friendship and support.

Finally, I am very grateful to my family who has always been there for me, and supporting me on my endless way to discover myself and life.

Martigny, September 2017 G.C.

Contents

List of Figures

List of Tables

1 Introduction

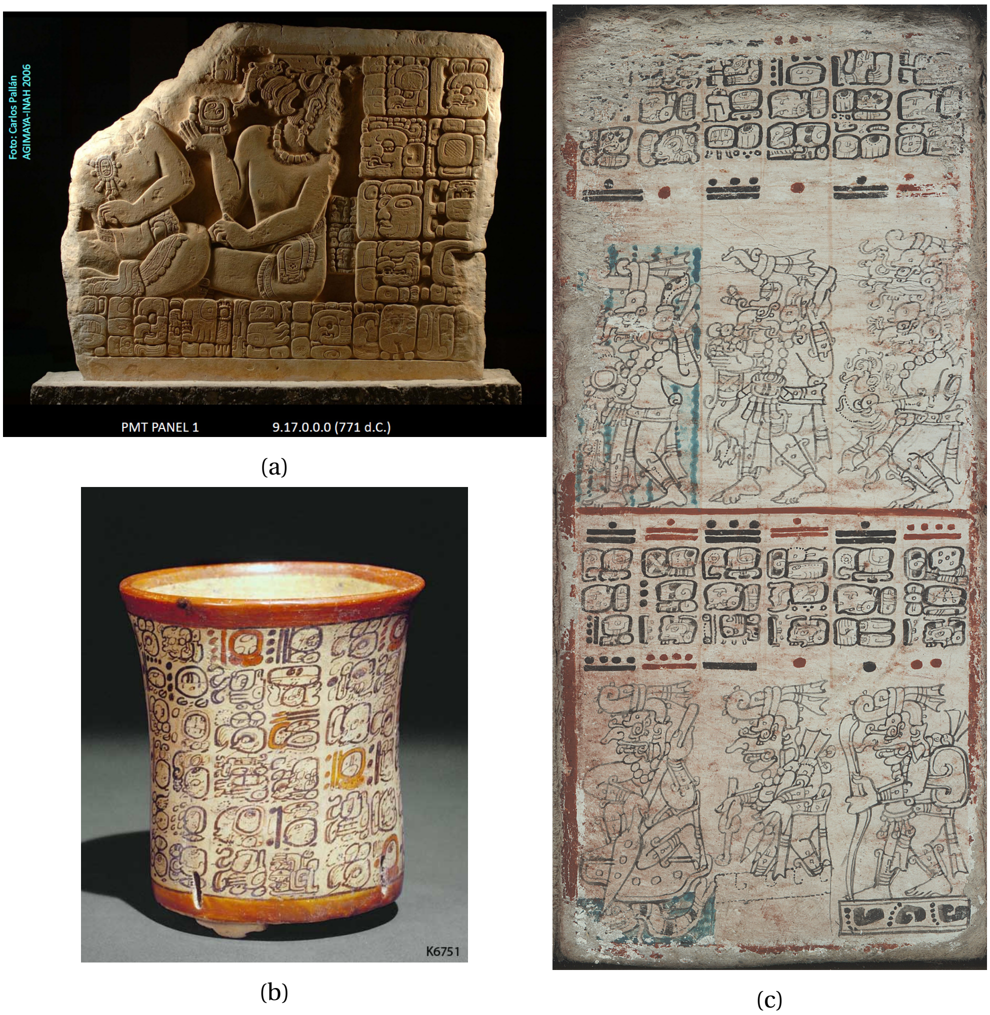
1.1 Motivation

Object recognition is an inherent talent of humans. For generations, humans observed their surroundings to survive and interact. This talent extended to recognition of abstract symbols with the invention of writing. Thanks to writing, humans were able to communicate on a common ground (in limited localities) and leave their cultural heritage to the next generations.

Without a doubt, the cultural heritage materials available to us today cannot make a complete picture. Natural disasters, wars, and degradation over time destructed a large amount of such materials. To prevent the further loss of cultural heritage materials, digitization, restoration, and other preservation techniques are essential. In the case of tangible heritage materials such as monuments or bark, this step can correspond to acquiring an image or a scan of the material, or reproduction. Usually, this step suffices for pure archival purposes or exhibitions in the museums. The next step is quest of understanding the context in which cultural heritage materials used, and their significance for the society.

Besides preservation and archiving, to pass the knowledge across generations, scholars face a challenging quest of understanding ancient writings from the available, incomplete data sources. To enable analysis of such cultural data sources, experts spend a significant amount of time for further processing after acquiring an image, e.g. cleaning raw images manually on available software, vectorizing, and annotating by checking available catalog sources. Since all the processing steps are repetitive and monotonous for experts, it is desirable to have alternative methods.

On the other hand, from the organization to the understanding of digitized cultural heritage materials, content-based analysis plays an important role. To analyze a historical piece, experts need to identify each element on the piece by mapping them to existing catalog sources, and to assess the relationship among the pieces (e.g. the reading order, the specific semantic, syllabic, or artistic roles, etc.). In the absence of catalog sources, experts may need to build a categorization system from scratch with the available data samples. Besides, analyzing the



(a)

(b)

(c)

Figure 1.1 – (a) A stone inscription found in Pomona, Tabasco (Mexico), Panel 1 from 771 AD, by courtesy of ©Carlos Pallán Gayol for AJIMAYA/INAH Project, 2006, Instituto Nacional de Antropología de Historia, Mexico. (b) A dynastic codex-style Maya vase[1] (K6751), by courtesy of ©Justin Kerr. (c) A high-resolution scan of page 65 from Dresden Codex[2] [The Saxon State and University Library, 1975] by courtesy of SLUB.

variations of the content might reveal important traces of the evolution of the language used across time periods and places, or differences in the artistic styles. In all the cases above, i.e. annotating signs (mapping to a category in a catalog), designing new catalogs, and observing the degree of sign variations in different sources, *quantitative assessment of the content* is crucial.

While the points mentioned above are generic in Digital Humanities, the focus of this thesis is on Maya hieroglyphs. The ancient Maya is one of the great civilizations in history, which has

[1] http://research.mayavase.com/kerrmaya_hires.php?vase=6751
[2] http://digital.slub-dresden.de/en/workview/?id=5363&tx_dlf

intrigued scholars worldwide, and produced a complex visual writing system. Maya writing is found on stone monuments, codex pages, and ceramic items (see Fig. 1.1). Maya writing contains glyphs that are grouped together in rectangular blocks. These glyph blocks are usually organized in a grid structure, and follow a “zigzag” (left to right, top to bottom) reading order. Apart from the text areas that contain glyphs, the writings may contain artistic elements such as icons.

With today’s know-how about the ancient Maya civilization, 80 percent of the glyphs are reported to be deciphered for their phonetic values. However, only around 60 percent of the glyphs are verified with respect to their meaning [Kettunen and Helmke, 2008]. The source of this difference comes from the cases where the phonetic value is known but the meaning is vague or not known and vice versa. Therefore, there is a strong driving force to invest in developing computer vision and multimedia techniques in order to help archaeologists to digitize, explore, and understand the ancient Maya writings.

1.2 Scope of the Thesis

One main research goal in this thesis is to design a crowdsourcing task to enable the contribution of non-experts to the above-mentioned, time-consuming preprocessing steps of data sources. The question is how much guidance is required during the task for non-experts to perform well. Even though expert knowledge cannot be replaced with few hints offered as part of a task design, for a well-defined task with enough guidance, we posit that it is possible to obtain an aggregated outcome of acceptable quality from the crowd. This thesis demonstrates this point in two crowd tasks with different amount of guidance, for the segmentation of glyphs out of blocks which is a task that to our knowledge was thought to be an endeavour that only experts could take on.

The second main research goal of the thesis is quantitative assessment of the content, in particular Maya glyphs. This thesis analyzes several knowledge-driven shape descriptors and investigates data-driven neural representations. Knowledge-driven descriptors are strong baselines in many computer vision and multimedia applications [Battiato et al., 2010; Belongie et al., 2002; Dalal and Triggs, 2005b; Del Bimbo and Pala, 1997; Eitz et al., 2010; Hu and Collomosse, 2013; Kazmi et al., 2013; Lowe, 1999; Mori et al., 2005; Yang et al., 2008]. The general idea of such descriptors is to design a filter bank and to represent an image, either globally or locally, as a composition of the frequencies of each filter. Such representations are pre-defined and fixed. Previously, a variety of global and local knowledge-driven shape descriptors were analyzed for Maya glyph retrieval for a limited amount of syllabic signs from monuments [Roman-Rangel et al., 2009, 2011a,b, 2013]. In this thesis, some of these shape descriptors are evaluated in different settings and at a larger-scale (both a larger number of categories and a larger overall dataset) for the Maya glyphs from the three known authentic codices (Dresden, Madrid, and Paris).

On the contrary, data-driven approaches aim to learn representations of an image directly from the data. One important point about the data-driven approach is the necessity of a considerable amount of data so that the algorithms can capture the patterns in the data. This point also emphasizes the necessity of the crowdsourcing tasks that are conducted in this thesis. Thanks to the second crowdsourcing task, a large-scale Maya codical glyph dataset is constructed and exploited via data-driven approaches such as convolutional neural networks. Furthermore, we study the characteristic parts of glyphs that enable archaeologists to determine the glyph category. We analyze whether such diagnostic parts of glyphs correspond to the discriminative regions according to the trained convolutional neural networks. In order to assess the strength of the learned neural representations qualitatively, and to observe the degree of correspondence between which regions experts find as diagnostic and what the networks learn as discriminative, we designed a crowdsourcing task. In this perspective, we analyze non-experts' perception on the performance of the trained networks.

1.3 Main Contributions

The contributions of the thesis, presented in the following chapters, can be summarized as follows:

1. **Exploring Local Shape Descriptors.** We investigated the local Histogram of Oriented Shape Context (HOOSC) shape descriptor [Roman-Rangel et al., 2011a] for the task of individual glyph classification. Specifically, we studied the descriptor in different settings, i.e. in a voting scheme rather than using a bag-of-words representation, with different spatial context, and with additional relative location information.

   Furthermore, we proposed a data-driven local shape representation based on Sparse Autoencoders (SA) [Hinton and Zemel, 1994]. We systematically evaluated the knowledge-driven HOOSC descriptor and our proposed SA representation on both Maya monument glyphs and a reference dataset of generic sketches. In this case, we also evaluated the HOOSC and SA with respect to different spatial contexts, and experimented with dense sampling of pivots for descriptor computation. The experiments show that the scale of data is important while learning neural representations, since the SA-based representation outperforms the HOOSC descriptor on the whole sketch dataset, however it stays behind the HOOSC's performance in the small amount of glyph data. Furthermore, we conclude that exploiting a large spatial context around descriptors is essential for shape data. Learning shape representations from raw data within a large spatial context requires a deeper network structure than the proposed single-layer SA-based representation so that shapes can be encoded more robustly and hierarchically.

2. **Crowdsourcing.** We collaborated with scholars in archaeology and epigraphy, in the context of the multi-disciplinary MAAYA SNSF project. Thanks to this collaboration, we had access to high-quality digital scans of the pages from the three authentic Maya codices. These scans were pre-processed, then segmented into blocks, and annotated by the collaborators. Thanks to this digitization and annotation process, we obtained a

glyph block dataset of around 7000 block images for automatic shape analysis.

For further processing, we proposed the use of crowdsourcing to generate a large-scale segmentations of *individual* Maya glyphs within blocks. The resulting unique dataset is composed of aggregated segmentations for over 9000 individual glyphs from the three authentic Maya codices. We experimented with different choices in the task design, and demonstrated that non-experts can produce segmentation of high-quality.

3. **Deep Neural Representations.** For the task of single glyph classification, we compared two knowledge-driven representations, namely HOOSC within a bag-of-words approach and Histogram of Gradients (HOG) [Dalal and Triggs, 2005b], as well as three data-driven representations by deep Convolutional Neural Networks (CNN), namely using the features from pretrained networks, transfer learning via fine-tuning of the last block of pretrained networks, and training networks from scratch with our crowdsourced individual Maya codical glyph dataset. The results show that the data-driven approaches outperform the traditional shape descriptors by a large margin on our individual Maya codical glyph dataset. Among the data-driven approaches, transfer learning approach improves over exploiting the pretrained CNN features; and training networks from scratch yields the best results. Furthermore, we observed that training a sequential sketch- specific network with few parameters from scratch with batch normalization [Ioffe and Szegedy, 2015], balanced oversampling, and dropout regularization [Hinton et al., 2012] performed slightly better than the recent residual models [He et al., 2016].

4. **Visualization.** We demonstrated the potential of locating the most discriminative/diagnostic parts of a glyph via Gradient-based Class Activation Map (GradCAM) method [Selvaraju et al., 2016]. This method is applied on the CNN models trained from scratch with our Maya codical glyph dataset. Furthermore, we applied this GradCAM method to detect candidate regions of individual glyphs from weakly-labeled glyph blocks. The results show that the trained model is mostly paying attention to diagnostic parts of the glyphs, and give clues on the candidate regions in the blocks even though visually-similar regions might also be marked.

   In order to assess the interpretability of the CNN visualizations, we conducted a crowdsourcing experiment to compare the expressivity of GradCAM method with two different trained models (a sketch-specific sequential convolutional network and a residual network). The results show that the sketch-specific network produces more appealing visualizations on the glyph diagnostic parts more often compared to the residual network.

   To index and visualize glyph shapes, we mapped a 10-class subset of the monument glyphs, the sign variants from the existing catalogs, a subset of the codical glyph-block images, and the crowd-segmented individual codical glyphs from high-dimensional feature space to two dimensional space via t-distributed Stochastic Neighborhood Embedding (t-SNE) [Van der Maaten and Hinton, 2008]. In the first case, we created the mapping on a traditional shape descriptor (the HOOSC-bag-of-words representation), whereas for the other cases, we focused on the representations from the pretrained CNNs. We discuss that this kind of indexing is valuable to understand the data clusters (that may correspond to different glyph variants in existing catalogs), to highlight the trends in the glyph shapes (i.e. gradual change in the style), and to investigate the similarity of unknown/vague glyph samples to other samples with known categories.

1.4 Outline

The thesis has 7 chapters. Chapter 2 introduces the Maya writing system, as well as the two catalog sources and the databases used in the thesis.

Chapter 3 presents the work on classification of Maya monumental glyphs with the HOOSC descriptor in different settings. This chapter also describes the comparative study between the knowledge-driven HOOSC descriptor and data-driven Sparse Autoencoders for glyph and sketch classification.

In Chapter 4, two crowdsourcing experiments are presented for glyph localization and segmentation tasks with differing amount of guidance provided to non-expert observers. Through the second crowdsourcing task, a large-scale Maya codical individual glyph corpus was curated.

In Chapter 5, for Maya codical glyph classification, several CNNs are systematically assessed in three settings: leveraging knowledge from pretrained networks, transferring knowledge from pretrained networks, and training a CNN from scratch. Furthermore, the CNNs trained with the glyphs were used to reveal the discriminative parts of the glyphs.

In Chapter 6, a crowdsourcing-based comparison of the visual activations from two CNNs is presented, which sheds some light on the interpretability of the deep networks. This chapter also includes indexing of glyph shapes that allows visual investigation and interpretations.

Finally, Chapter 7 concludes the thesis by reflecting upon this dissertation's contributions, limitations, and possible further research directions.

1.5 List of Publications

This section presents the related publications that are outcomes of our research.

1.5.1 Journals

- Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Maya codical glyph segmentation: A crowdsourcing approach. Research Report Idiap-RR-01-2017, Idiap, January 2017c (accepted for IEEE Transactions on Multimedia)

- Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Evaluating shape representations for Maya glyph classification. *ACM Journal on Computing and Cultural Heritage*, 9 (3), Sep 2016a

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

1.5.2 Conferences and Workshops

- Gülcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Shape representations for maya codical glyphs: Knowledge-driven or deep? In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, CBMI '17, pages 32:1–32:6, New York, NY, USA, 2017a. ACM. ISBN 978-1-4503-5333-5. doi: 10.1145/3095713.3095746. URL http://bib-ezproxy.epfl.ch:2512/10.1145/3095713.3095746

- Edgar Roman-Rangel, Gulcan Can, Stephane Marchand-Maillet, Rui Hu, Carlos Pallan Gayol, Guido Krempel, Jakub Spotak, Jean-Marc Odobez, and Daniel Gatica-Perez. Transferring neural representations for low-dimensional indexing of Maya hieroglyphic art. In *ECCV Workshop on Computer Vision for Art Analysis*, October 2016

- Gulcan Can, Jean-Marc Odobez, Carlos Pallan Gayol, and Daniel Gatica-Perez. Ancient Maya writings as high-dimensional data: a visualization approach. In *Digital Humanities*, 2016b

- Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Is that a jaguar? Segmenting ancient Maya glyphs via crowdsourcing. In *ACM International Workshop on Crowdsourcing for Multimedia*, pages 37–40. ACM New York, November 2014. doi: 10.1145/2660114.2660117

1.5.3 Reports

- Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. How to tell ancient signs apart? Recognizing Maya glyphs with CNNs. Idiap-RR Idiap-Internal-RR-26-2017, Idiap, April 2017b (planned to be submitted to ACM Journal on Computing and Cultural Heritage)

2 The Maya Writing System

Humankind started to leave traces of their lifestyle, beliefs, and philosophy thanks to writing systems. Writing can be seen as an act of drawing certain shapes in a predefined order. In some ancient semanto-phonetic systems, e.g. Egyptian, Luwian, and Maya writings, these shapes were sketch-like and resembled natural objects, scenes, events (astronomical, royal or religious), and encounters with animals and humans. Among these writing systems, the Maya writing is still under a decipherment phase and intrigues many scholars.

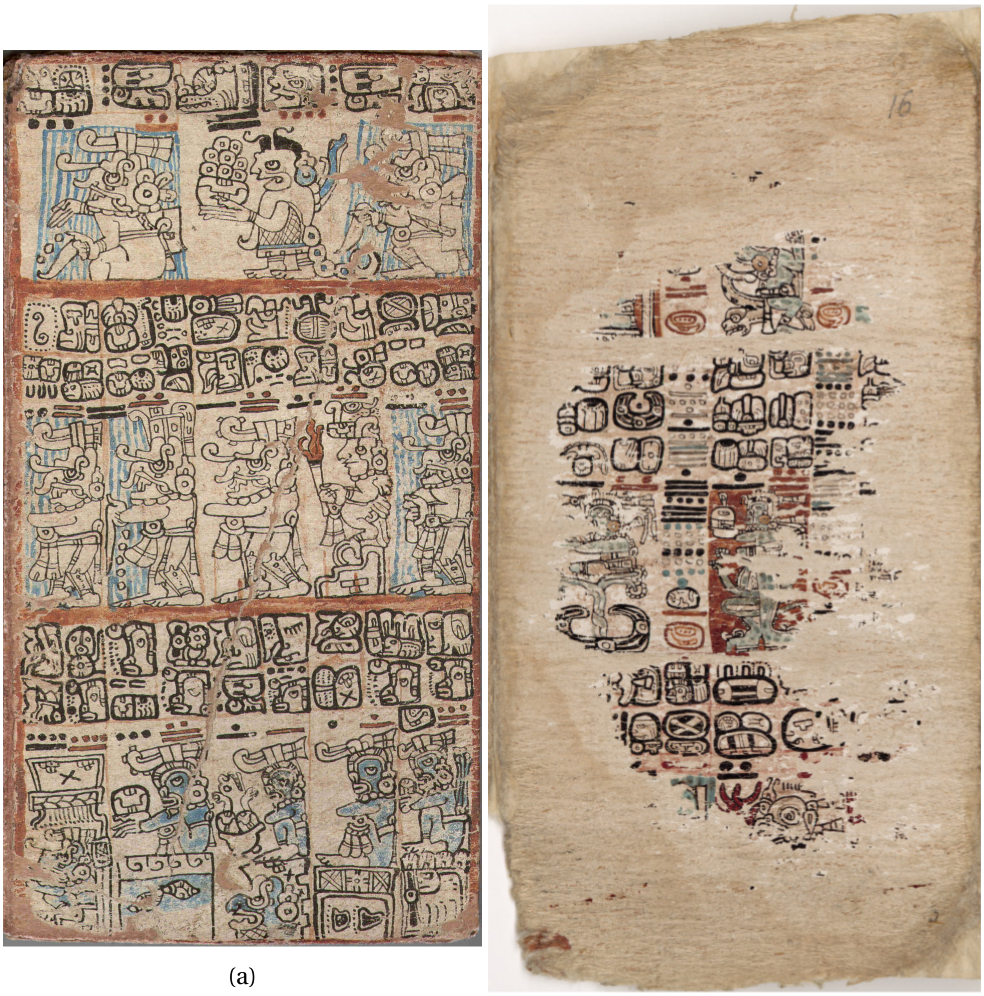
This chapter gives a brief introduction to the Maya civilization, the Maya writing system, existing catalog resources, and the Maya data that is used throughout this thesis.

2.1 The Maya Civilization

The ancient Maya civilization flourished between 2000 BC to 1600 AD in Mesoamerica, more specifically in Mexico, Guatemala, Belize, and Honduras.

Among all the civilizations in Mesoamerica, such as the Olmecs, Teotihuacans, Aztecs, and many others, the specific writing system of the Maya made this civilization distinct [Miller and Taube, 1993]. Thanks to their writing system, Maya people were able to leave their mark across places and time periods. The prominent Maya cities had been populated across the preclassic (2000 BC – 250 AD), classic (250 – 900 AD), and postclassic periods (950 – 1539 AD). The period between 1511 and 1697 AD is referred as contact period. During this period, many cultural heritage materials were destroyed by Spanish conquerors.

The Mayas left a great amount of cultural heritage materials, such as stone inscriptions (Fig. 1.1a), carved wood monuments, folded bark pages (called as *codex*, e.g. Fig. 1.1c and Fig. 2.1), or ceramic items (Fig. 1.1b). The common ground of all these materials are the hieroglyphs, in short *glyphs*, written on them.



(a)

(b)

Figure 2.1 – High resolution scans of sample pages from two genuine Maya codices (see Fig. 1.1c for a sample page from Dresden Codex). (a) Page 11 from Madrid Codex [Museo de América, 1967]. (b) Page 16 from Paris Codex [Bibliotheque Nationale].

2.2 The Maya Writing System

A *glyph* is a unit sign of the Maya writing. A *glyph-block* is composed of several glyphs. A *glyph text area* comprises of many glyph-blocks that are structured as a grid (see Fig. 2.2b), and has a specific reading order in general (top to bottom, and left to right in double columns). A typical Maya artifact (a stone inscription, a codex page, or a ceramic item) is composed of text areas, and some other pictorial elements such as icons or calendrical signs (see Fig. 2.2a). A *t'ol* is a chapter-like unit that is separated by colored lines in a codex page. A *codex* is composed of several pages with the sub-elements described above.

The Japanese language has Hiragana alphabet for syllabic (pronunciation) and Kanji alphabet for logograhic (semantic) significance. Similar to the Japanese language, the Maya language also has a syllabo-logographic or semanto-phonetic writing system. Fig. 2.2a illustrates examples of syllabic and logograhic glyphs. Logograhic glyphs may have syllabic forms, and due to space limitations or aesthetic reasons, they might be used interchangeably. For instance, the logographic version of glyph T229 (Thompson code, see Section 2.3) in the second row of Fig. 2.3 is an impersonated form (resembling a head) of the syllabic versions next to it. Furthermore, in a Maya text, some syllabic glyphs with the same pronunciation may be used in place of each other. These observations enabled the decipherment of the frequently-occurring Maya glyphs [Kettunen and Helmke, 2008].

2.2.1 Variations in Glyph Signs

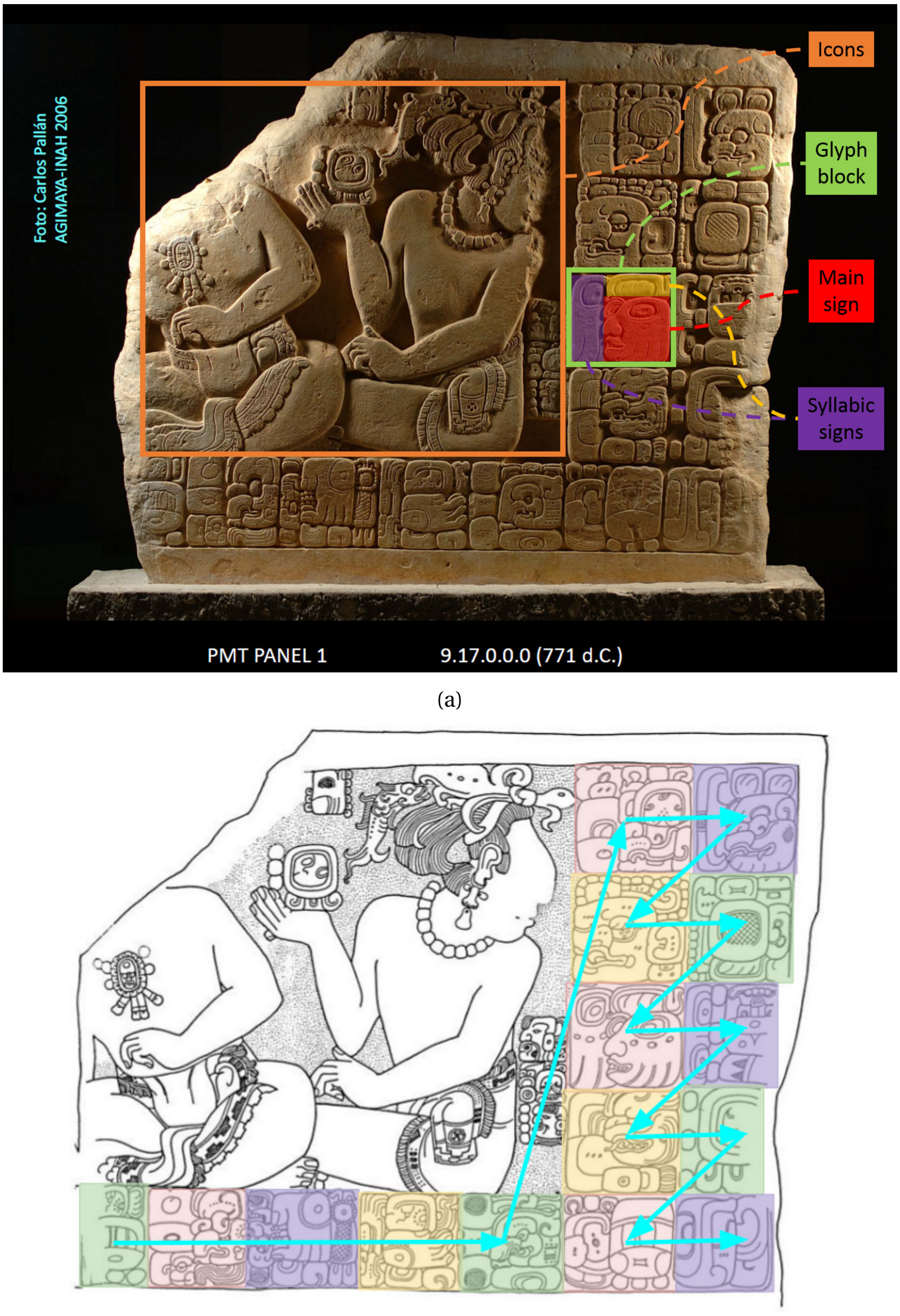
A writing system evolves throughout time like a living being. Besides the changing nature of languages due to abandoned or nontransferable knowledge to next generations, encounters and interactions with different cultures impact a language and its writing system.

Furthermore, written texts may also vary due to artistic reasons, space limitations, and different materials and means that are used, i.e. carving stones with a metal tool, or painting on ceramic/bark with a brush. These variations might be observed as the differing amount of details, deformations on the general form of the signs, size variations, and compound signs that contain only the characteristic parts of several signs (see Fig. 2.3).

Fig. 1.1c and Fig. 2.1 illustrate the stylistic differences in the three existing Maya codices. Although all the codices are thought to be reproduced from their older versions in the 15^{th} century (post-classical period) [Kettunen and Helmke, 2008], the glyphs in the Madrid Codex are larger and more rectangular than the instances in the two other codices.

Maya glyphs from one category may look relatively different to the exception of some characteristic parts [Houston et al., 2000]. Fig.2.4 shows several variations of the “Chaa'k” Rain God (T0668, MZ9) glyph in the codices. These examples illustrate the deformations in the general form (outer contour) as well as missing or extra inner elements, such as the “thumb”

[2] http://research.famsi.org/uploads/schele/hires/11/SD7628.jpg



(a)

(b)

Figure 2.2 – (a) The main elements in Maya writing illustrated on the stone inscription from Fig. 1.1a. (b) A digitized drawing of the stone inscription Schele et al. [1986] by courtesy of ©Schele. This partial drawing is adopted from the FAMSI website[2]. The glyph-blocks in the text area are illustrated (Blue arrows indicate the reading order).

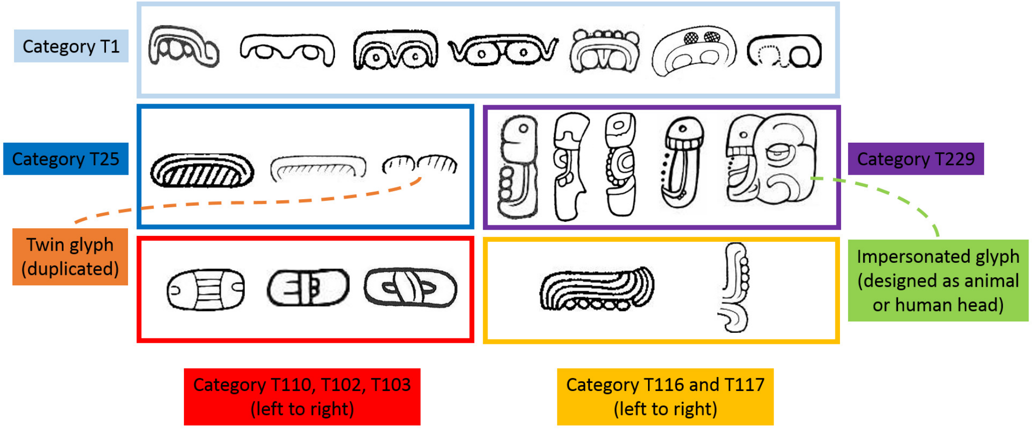

Category T1
Category T25
Category T229
Twin glyph (duplicated)
Impersonated glyph (designed as animal or human head)
Category T110, T102, T103 (left to right)
Category T116 and T117 (left to right)

Figure 2.3 – Selected Maya glyph samples from several categories that illustrate the within-class variety (first two rows) and between-class similarity (last row). The first two rows show samples from the same category in each box. The last row shows visually similar samples from different categories. See the text for details.

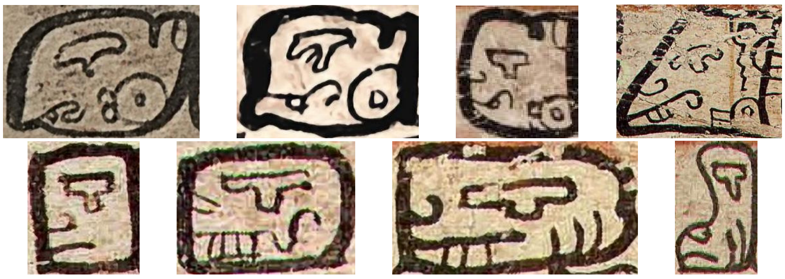


Figure 2.4 – The variations in the codical "Chaa'k" (Rain God, MZ9) glyph sign.

on the top right or the two circular elements (one inside the other) on the bottom right of the glyphs in top row. Please notice that the “nose” or the “teeth on the mouth” may not be visible in some samples. The hollow “T-eye” shape (or “ik” sign) enables the experts to label these samples as belonging to the same category. In the following chapters, such characteristic parts of glyphs are referred as *diagnostic* parts.

2.3 Catalog Sources

The documentation of Maya writing started during the Spanish conquest of Yucatan in the XVI^{th} century. Bishop Diego de Landa’s first incomplete alphabet, in his book titled “Relación de las cosas de Yucatán” [de Bourbourg, 1864; Tozzer, 1941], was created by asking two locals how to write Spanish characters in Maya language [Wikipedia, 2016]. Later on, for several centuries, few attempts were made to understand Mayan writings. [Evrenov et al., 1961]’s and [Thompson and Stuart, 1962]’s sign catalogs became important sources, suggesting syllabic readings rather than character correspondences of the signs. For historical reasons, Thompson’s taxonomy (main and affix syllabic signs) became more influential than Evrenov’s for several decades. With the advancement of the understanding of the semantics of the signs, more modern catalogs emerged [Macri and Looper, 2003; Macri and Vail, 2008].

As illustrated in Fig. 2.5, the Thompson catalog has three main categories: affix, main, and portrait signs. The Macri-Vail taxonomy has 13 main categories [Macri and Vail, 2008] which is encoded in the first digit of three-digit encoding of categories. Six of them, i.e. animals, birds, body parts, hands, human faces, and supernatural faces, are grouped semantically (according to. the visual elements of the most-frequent first-listed variant). There is a main category for numerical signs that are composed of dots and bars. The rest are grouped based on visual elements (square signs divided based on symmetry, and elongated signs divided based on the number of components). The second digit in the Macri-Vail is based on the defined visual cues, and the third digit is simply for numbering. Macri and Looper discuss that some semantic signs (e.g. hand signs) are spread over different categories in the Thompson catalog [Macri and Looper, 2003] (see Fig. 2.6). Furthermore, the ordering or the distinction of some Thompson signs is argued not to follow a clear rule.

Since Thompson’s catalog was highly adopted for a long time and Macri-Vail’s catalog has a modern taxonomy with a focus on Codices signs, we use these two resources. The fundamental difference between them is the emphasis given to visual appearance and to semantics. Thompson is known to categorize the glyphs with respect to similarity based on hand-prepared graphic cards. Macri-Vail consider co-occurrences of the signs and modern knowledge of the semantics and usage of some signs rather than visual cues only. This leads to a higher within-class visual dissimilarity of Macri-Vail signs. For instance, as shown in Fig. 2.6, the variants in the AMB category are spread over three Thompson categories (T534 main sign, T140 and T178 affix signs).

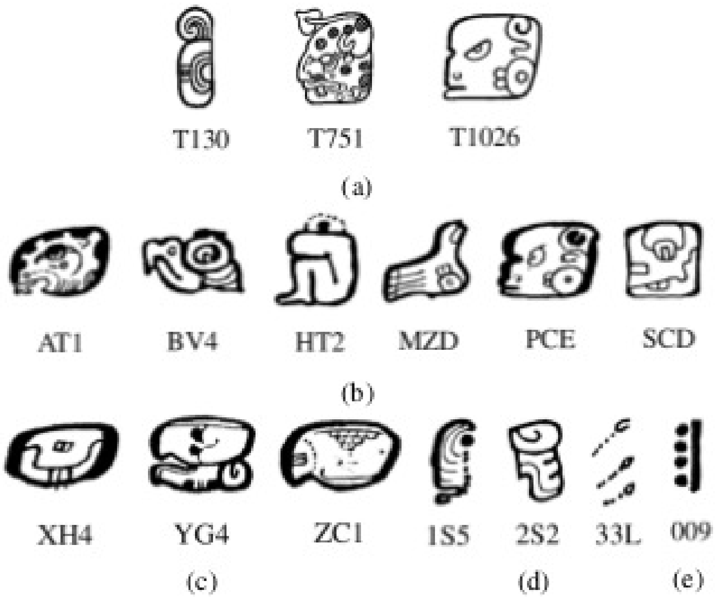


Figure 2.5 – (a) Examples for affix, main, and portrait categories in Thompson and Stuart [1962]. Last two rows illustrates examples for main categories in Macri and Vail [2008]: (b) semantic (animals, body parts, and faces); (c) square (symmetric, asymmetric, and with irregular shapes); (d) elongated (with 1, 2, and 3 components); and (e) numeric categories.

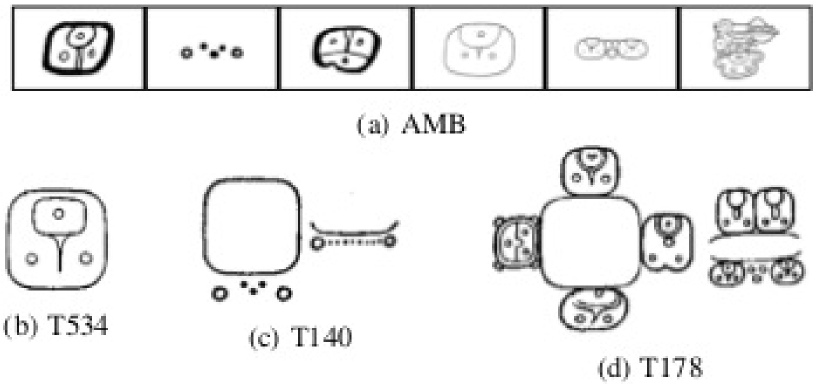


Figure 2.6 – (a) Variants of AMB category in Macri and Vail [2008] catalog; (b-d) occurrences of these variants in 3 different categories in the Thompson and Stuart [1962] catalog.

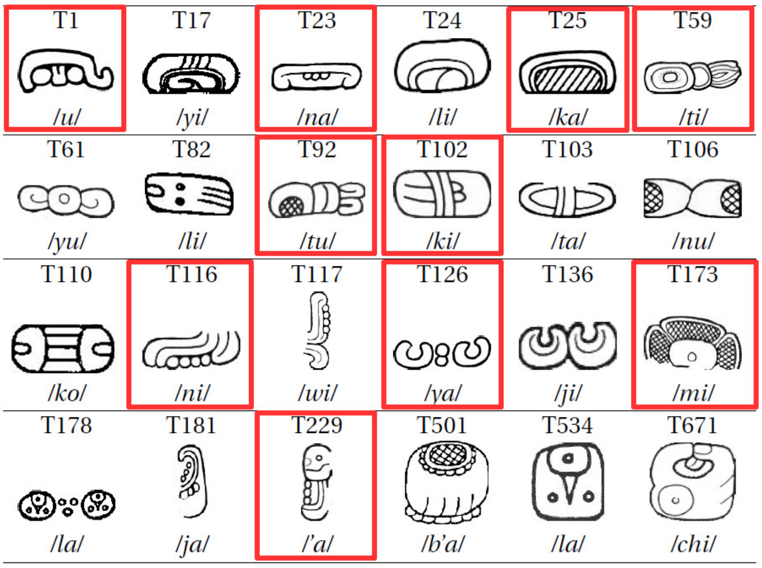


Figure 2.7 – Sample glyph images, corresponding Thompson annotations and syllabic values (sounds) from 24-class syllabic Maya glyph dataset, by courtesy of Carlos Pallán Gayol for the ©INAH/CODICES project. The selected 10 classes (used in Chapter 2 and 6.2.4) are outlined with red.

2.4 Monument Data

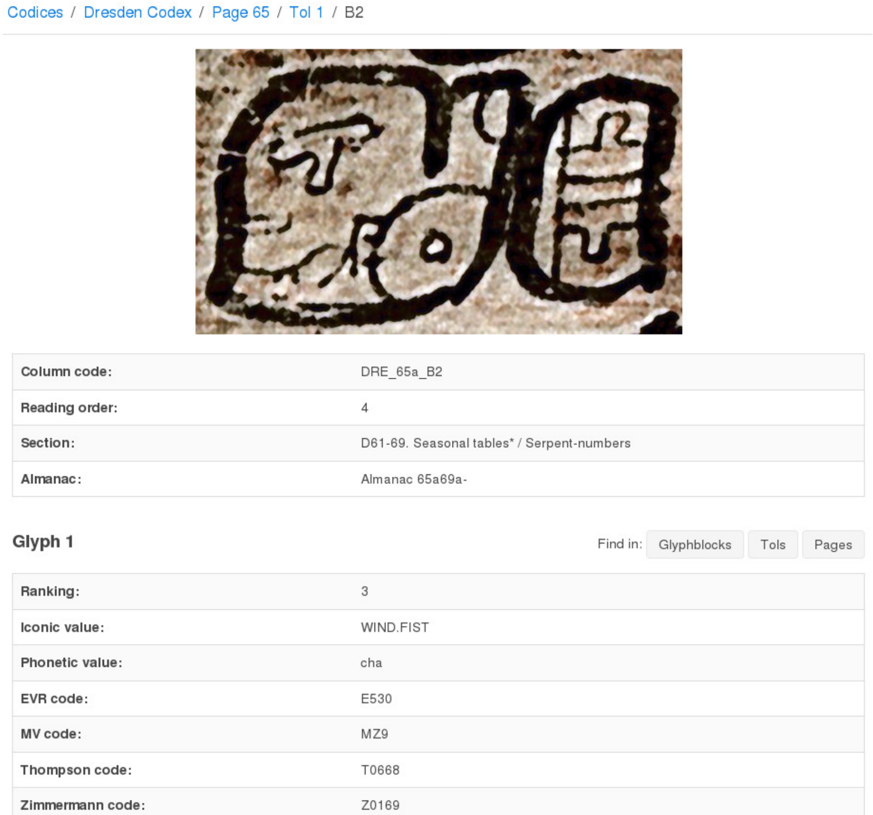
Experts in archaeology and epigraphy digitize monument inscriptions by taking the photographs under special lighting conditions. In this way, the details of the carvings are captured. Then, experts produce a binary drawing based on the photographs. It is non-trivial to produce this type of data. Epigraphers follow strict quality and fidelity standards to generate hand-drawn glyphs from the original sources.

The experiments in Chapter 3 is based on this kind of monument glyph data. The monument dataset [Roman-Rangel et al., 2011a] contains 24 classes of frequent syllabic glyphs inscripted on monuments, which were collected from four subregions of the Maya area, i.e., Petén, Usumacinta, Motagua, and Yucatán.

As an additional source, around 300 glyph samples are taken from existing catalogs like Macri and Looper [2003], and Thompson and Stuart [1962]. Since this dataset is not balanced (i.e. several glyph categories have very few examples), we only kept the 10 classes with the largest number of examples (marked with red in Fig. 2.7) for the classification experiments in Chapter 3.

Table 2.1 – Number of elements in the three codices.

	# pages	# blocks	# glyphs	# glyphs with annotation and source image
Dresden	72	2924	6932	6439
Madrid	100	3254	7429	6910
Paris	18	774	1620	1373
All	190	6952	15981	14722



Codices / Dresden Codex / Page 65 / Tol 1 / B2

Column code:	DRE_65a_B2
Reading order:	4
Section:	D61-69. Seasonal tables* / Serpent-numbers
Almanac:	Almanac 65a69a-

Glyph 1

Ranking:	3
Iconic value:	WIND.FIST
Phonetic value:	cha
EVR code:	E530
MV code:	MZ9
Thompson code:	T0668
Zimmermann code:	Z0169

Figure 2.8 – A sample metadata page for a glyph-block in the first t'ol of 65^{th} page in Dresden Codex. This page is part of the database that is organized and stored by our project partners in University of Geneva for the MAAYA project. http://maaya.unige.ch/codices/1/pages/44/tols/148/glyphblocks/330

Table 2.2 – Number of individual glyphs with different quality rating (0-4).

	0	**1**	**2**	**3**	**4**
Dresden	304	184	493	2680	3229
Madrid	223	505	924	1974	3766
Paris	261	20	222	371	667
All	788	709	1639	5025	7662

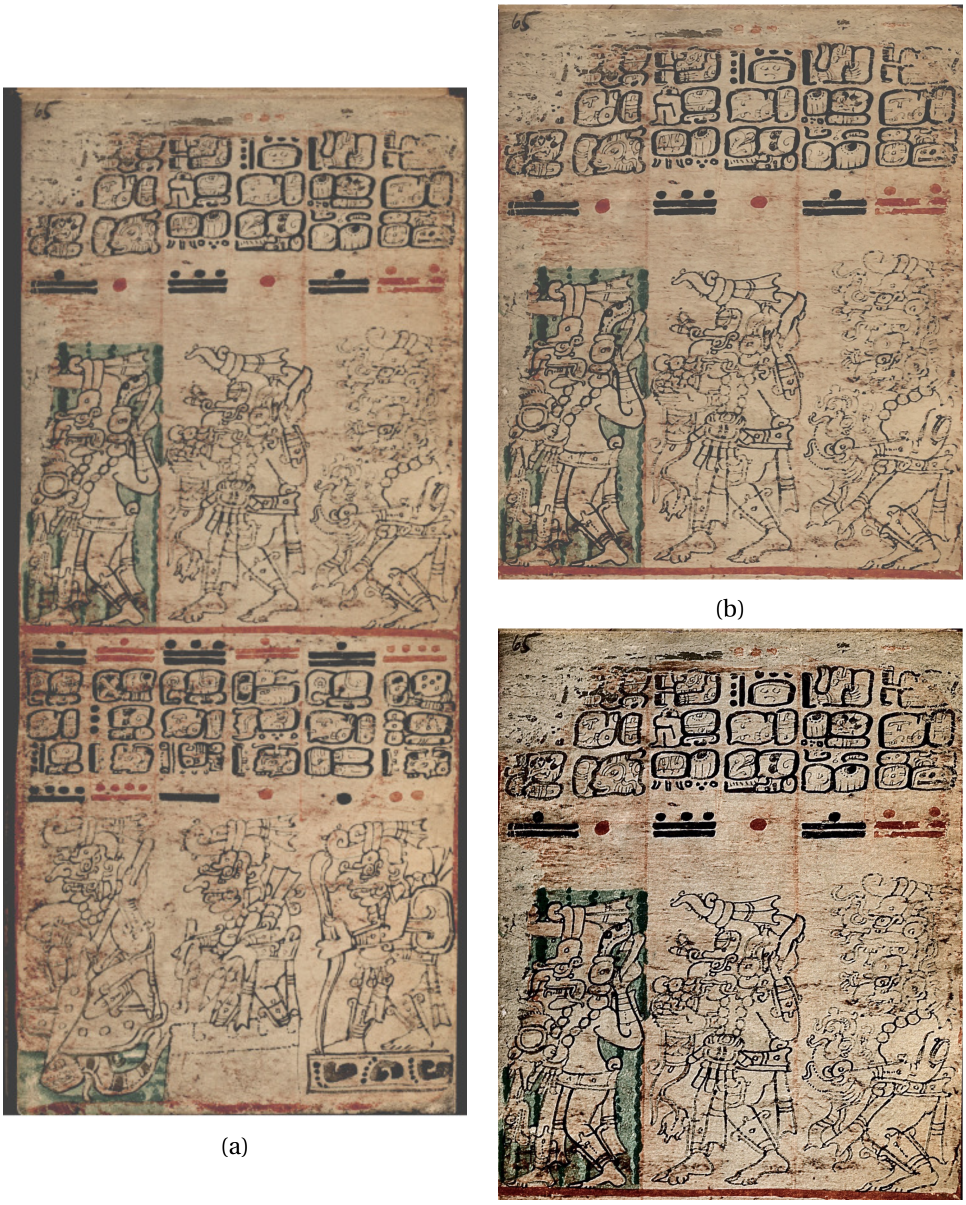
2.5 Codices Data

A key source used in this thesis is the high-resolution digital images from the three existing Codices (Dresden [The Saxon State and University Library, 1975], Madrid [Museo de América, 1967], and Paris [Bibliotheque Nationale]), cropped to smaller units (pages, t'ols, and glyph-blocks), and annotated with metadata. Images and annotations were all provided by project partners in epigraphy. Specifically, Carlos Pallán Gayol (Uni. of Bonn), Guido Krempel (Uni. of Bonn), and Jacub Spotak (Comenius Uni. in Bratislava) produced the data from Dresden Codex, Madrid Codex, and Paris Codex respectively. Our project partners in University of Geneva organized and stored all the information (i.e. images, annotations, and metadata) in a database. Fig. 2.8 illustrates a sample page from the database. We obtained the data to use in our experiments through this database.

Table 2.1 summarizes the number of elements available from the three Codices. Some pages of the Codices are highly damaged. Even though there are, respectively, 76, 112, and 22 pages in our database, we only list the number of pages that have at least one recognizable glyph in Table 2.1. Similarly, we have the records of 7047 glyph-blocks in total, however only 6952 of them have at least one recognizable glyph. In total, 14722 glyphs have known catalog annotations with cropped glyph-block images.

The metadata of each glyph-block contains the name of the codex, page number, t'ol number, reading order, and relative location (row and column order in the t'ol). The metadata of each glyph in each glyph-block contains its reading order, its sign code from various catalogs ([Thompson and Stuart, 1962], [Macri and Vail, 2008], [Evrenov et al., 1961], and [Zimmerman, 1956]), its phonetic value, and its damage level. The damage level is rating by the expert in a range from 0 (undecipherable) to 4 (high quality), and indicates how identifiable the glyph is. This is not decided only based on visual degradation, but also based on the semantics and co-occurrence with neighboring glyphs. For instance, a glyph with a damage rating 2 is degraded and misses certain parts, but it is still identifiable based on the remaining strokes/parts and the context of the usage. Table 2.2 indicates the number of glyphs from each codex according to their damage level. This table shows that 80 % of the glyphs are good or very good conditions (rated 3 or 4). In Chapter 4 and 5, we based our experiments on such glyphs that are in good conditions.

Preprocessing. The experts (project partners in epigraphy) provided the codices data after



(a)

(b)

(c)

Figure 2.9 – (a) Color correction of the full page (page 65 in the Dresden Codex), (b) cropping and upsampling the *t'ol a* (top chapter-unit that is separated by a red line on the page), (c) unsharpening the upsampled t'ol. These images were produced by Carlos Pallán Gayol for the ©MAAYA project.

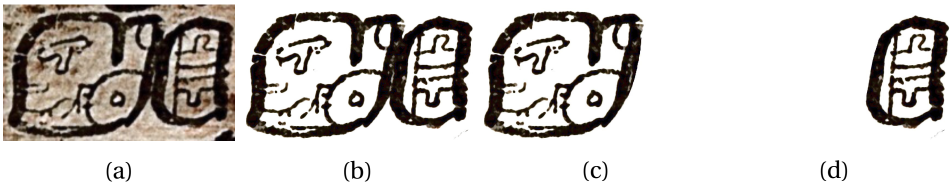


Figure 2.10 – (a) Cropping a glyph-block, (b) binarization of the glyph-block, (c-d) cropping the first and second glyphs. These images were produced by Carlos Pallán Gayol for the ©MAAYA project.

preprocessing with commercial tools. The details of these preprocessing steps are as follows:

1. A color correction was applied on a full codex page (Fig. 2.9a).
2. Each t'ol (chapter-like unit separated by a colored line) on the page were cropped and upscaled to four time of its actual size (Fig. 2.9b).
3. An unsharpening process was applied on each t'ol (Fig. 2.9c).
4. Each glyph-block in the text area of the t'ol was cropped (Fig. 2.10a).
5. Each cropped block was binarized and cleaned to eliminate the background color and the strokes from the neighboring blocks (Fig. 2.10b).
6. Finally, each glyph in the block was cropped, i.e., the strokes belonging to the other glyphs were eliminated (Fig. 2.10c-2.10d).

Note that the epigraphy experts have not provided the clean block images and the individual glyph images for all the instances, as cleaning the blocks and the segmentation of blocks into individual glyphs is demanding in terms of time and effort. Furthermore, deciding annotations of glyphs for several catalogs, assigning identifiability rating, and providing spellings are quite time-consuming. As the experts' focus is on decipherment, only a very small proportion of individual glyph segmentations (341 from the Dresden Codex, 302 from the Madrid Codex, and 116 from the Paris Codex) were previously produced by them as in [Hu et al., 2015]. At the large scale, the experts provided only the cropped block images (as in Fig. 2.10a) without binarization or cleaning. Therefore, in order to obtain the individual glyph regions in the blocks, we designed a segmentation-oriented crowdsourcing task. This process is presented in detail in Chapter 4.

2.6 Conclusion

In this chapter, we briefly introduced the Maya writing system, the challenges that are inherent in the Maya data such as the glyph variations. Then, we explained the structure of the existing prominent Maya glyph catalogs. Finally, we described the data sources that are used throughout this thesis. Specifically, the syllabic glyph data from monument inscriptions was studied in Chapter 3, whereas all the remaining chapters focus on the codices data.

3 Local Shape Representations

In this chapter, we describe a study of local shape representations on binary shape data. We investigate both a knowledge-driven traditional shape descriptor (HOOSC) and a data-driven representation based on Sparse Autoencoders (SA). After reminding the motivation for the shape classification task, we introduce the methodology, give the specific information about the datasets used in this chapter, present and discuss the results, and draw conclusions.

The contributions presented in this chapter was originally published in the following journal paper[1] :

- Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Evaluating shape representations for Maya glyph classification. *ACM Journal on Computing and Cultural Heritage*, 9 (3), Sep 2016a

3.1 Introduction

In this thesis, we collaborate with archaeologists and epigraphers to define potential use cases and work towards glyph classification tools. For instance, for training purposes, when a novice archaeologist sketches a glyph, the system could provide the most likely categories that this glyph would belong to. Another potential use case is that the archaeologist could take a photograph of a glyph instance from a stone monument or folded codex and ask the system to categorize this instance. If the visual categorization system is reliable, the archaeologist would not need to search through the whole existing catalog to annotate the glyph instance. For damaged glyph instances, even though the opinions of archaeologists may differ due to the subjectivity of visual appearance, the system could be utilized by the archaeologists to get an additional quantitative assessment of visual similarity. Furthermore, for archaeologists, visual details at different scales are important to recognize a glyph instance. This points towards the necessity to study visual representations at different spatial contexts. Moreover, the glyph

[1] A related study (presented in Appendix A) about the settings of the traditional HOOSC descriptor was published as a part of a journal paper [Hu et al., 2015].

instances in a category may exhibit many variations, as shown in Fig. 2.3 in Section 2.2.1, yet be consistent with respect to certain diagnostic details. In our methodology, a bag-of-words representation is a possible way to address this issue, as the shared diagnostic details of the glyph instances from the same category may correspond to specific bins in the histogram representation after clustering the patch-based local representations. Our study contributes to an essential component of visual categorization systems, namely how to represent the visual appearance of the glyphs.

This motivates the study of visual representations that accurately discriminate glyph categories, and in general arbitrary shapes. The focus within this chapter is the systematic comparative analysis of knowledge-driven visual descriptors and data-driven representations for binary shape datasets. Specifically, the knowledge-driven representation is the previously proposed HOOSC descriptor for Maya glyphs [Roman-Rangel et al., 2011a] and the data-driven shape representation is learned by a single-hidden-layer sparse autoencoder [Hinton and Zemel, 1994]. The comparison is conducted through the impact of representations on classification performance.

On the one hand, HOOSC is a local descriptor that summarizes the shape regions as frequency histograms of line orientations. HOOSC is similar to SIFT [Lowe, 1999] and HOG [Dalal and Triggs, 2005b], which are commonly-applied in the object recognition literature. HOOSC is preferred in this study due to its competitive performance in hieroglyphic representation tasks [Franken and van Gemert, 2013; Roman-Rangel, 2012].

On the other hand, sparse autoencoder representations have attracted attention in computer vision due to their ability to capture hidden structures in the data in an unsupervised manner. The learned representation is expected to be a non-linear mapping that would cover edges in different angles and translations. Such data-driven representations may be more beneficial to capture data-specific patterns. Key research questions are how generic these representations are, and how well such representations perform compared to established handcrafted descriptors in the shape representation task.

Note that we focus on single-hidden-layer sparse autoencoders, since it is worth understanding the limits of representation learning with “shallow” networks in comparison to hand-designed descriptors. Alternative “deep” networks such as Convolutional Neural Networks (CNN) with many layers are investigated in Chapter 5.

In this chapter, we investigate the effects of model parameters of single-hidden-layer sparse autoencoders (including regularization parameters, number of hidden units), and the scale of the dataset on the classification results, and whether they are able to give comparable results to traditional descriptors. In this respect, we quantify how effective and generic each of these representations are on a Maya monument dataset as well as on a much larger human sketch dataset.

Contributions and plan: The contributions of this chapter are the following:

1. the evaluation of the performance of a data-driven auto-encoder approach for shape representation;
2. a comparative study of hand-designed HOOSC and data-driven SA, which to our knowledge has not been conducted previously;
3. an experimental protocol to assess the effect of the different parameters of both representations;
4. the creation of a bridge between computer vision and machine learning and Maya studies, specifically for visual analysis of glyphs.

From our experiments, the main conclusions are that the data-driven representation performs overall in par with the hand-designed representation for similar locality sizes on which the descriptor is computed. We also observe that a larger number of hidden units, the use of average pooling, and a larger training data size in the SA representation all improved the descriptor performance. Additionally, the characteristics of the data and stroke size plays an important role in the learned representation.

The rest of the chapter is organized in five sections. Section 3.2 presents the related work on object recognition and discusses some of the representations in computer vision for this task. Section 3.3 explains the classification pipeline used in this chapter, and describes the two studied representations. Section 3.4 summarizes the experimental setup, dataset details, and parameter settings. In Section 3.5, the results are discussed with respect to certain aspects and parameters of the pipeline. Finally, Section 3.6 concludes the chapter by pointing out the potential and limitations of both representations.

3.2 Related Work

This section presents the related work under the three main points: visual representations for object recognition, knowledge-driven representations for shapes, and data-driven representations.

Object representation and pooling. Object recognition architectures are generally composed of three steps: low-level feature extraction, coding, and pooling. Low-level features of an image, such as color, intensity values, or orientation of the edges, can be extracted locally over windows sliding through the image or over a defined region. The coding step relies on the availability of a dictionary. It is often obtained by clustering low-level feature vectors of all images. The dictionary is a more abstract representation, aiming to group and characterize repetitive patterns in the features across images. Given a dictionary, low-level local features of each image are coded, for instance, by quantizing them to their nearest element in the dictionary or by extracting their sparse decomposition with respect to the dictionary vectors. Finally, in the pooling step, the coded features are combined spatially (e.g. through averaging or by taking the maximum value) and a global representation is obtained for the whole image.

In the last decades, different modules of this common pipeline have been investigated. Boureau et al. [2010] use the Caltech-101 object recognition and 15-scene recognition datasets as benchmarks. They utilize dense SIFT [Lowe, 1999] as basic image features and propose “macro-features” in which neighboring features are encoded jointly. Macro-features improve only the object recognition results. In another similar study, Chatfield et al. [2011] analyzes feature encoding methods for bag-of-words image representations (BoW) [Sivic and Zisserman, 2003b] in object classification tasks for the PASCAL VOC and Caltech-101 datasets. The selected methods are locality-constrained linear encoding (LCL) [Wang et al., 2010], improved Fisher vector (FV) [Perronnin et al., 2010], super vector (SV) encoding [Zhou et al., 2010] and kernel codebook (KC) encoding [van Gemert et al., 2008]. Compared to the traditional hard quantization, the above methods keep more information about the original feature vector. Fisher vector coding is reported to give superior results. Furthermore, in the empirical analysis, both larger vocabulary size and higher sampling density are noted as critical aspects for good performance.

There are several common practices in the literature for pooling local descriptors and obtaining a global image representation. Spatial pyramid matching (SPM) [Lazebnik et al., 2006] is a popular approach to incorporate spatial information into BoW representations. This approach treats the image on several spatial levels. For each level, it divides the image into a grid of different granularity, i.e. level-0 (whole image), level-1 (2x2 grid), and level-2 (4x4 grid). Then, the BoWs in each grid cell of each level are normalized according to the cell area and stacked together. The SPM approach might not be optimal for some object classes. Thus, instead of pre-defined regular grids, it has been proposed to learn the spatial cells from an over-complete set of arbitrary rectangles over the image and call these cells receptive fields [Jia et al., 2012]. In a similar study, Battiato et al. [2010] analyzes hierarchical bag-of-words for scene classification. Each scene is partitioned into subregions hierarchically and described via texton distributions, which yields good performance for this task.

However, in general, the basic way to operate pooling is by taking the sum or max of the features in the defined spatial neighborhood. In the case of deep neural networks, the spatial neighborhoods for pooling can be densely and regularly sampled over the image. As an alternative, the spatial neighborhoods can be simply defined around specific points of interest. Since the main interest of this chapter is shape data, we use the latter and sample features around pivot points, which are points from the shape.

Knowledge-driven representations for shapes. In the literature, gradient-based handcrafted features such as SIFT [Lowe, 1999] and HOG [Dalal and Triggs, 2005b], and their variants [Eitz et al., 2012b; Roman-Rangel et al., 2011a] are popular and have been shown to work well in image recognition tasks. Roman-Rangel [2012] proposed the HOOSC shape descriptor, which is a combination of the HOG and Shape Context (SC) features, for shape retrieval tasks applied to Maya glyphs, ancient Chinese writing, and MPEG-7 shape dataset. In a more recent study, we showed that the performance can be boosted by adding the relative position of the pivot points where the HOOSC descriptors are computed [Hu et al., 2015]. Note that the details of this study are presented in the Appendix A. Furthermore, Roman-Rangel et al. [2012] evaluated hard vector quantization and sparse coding methods and reported that hard quantization is superior to sparse coding approach with l_1 norm on a specific Maya glyph dataset from monuments. In a similar hieroglyph recognition task for Egyptian writing, Franken and van Gemert [2013] evaluated HOG, SC [Belongie et al., 2000], and HOOSC [Roman-Rangel et al., 2011a]. All three descriptors were used with a bag-of-words representation, and performed similarly. By introducing spatial matching of the interest points, the results using HOG and HOOSC are further improved.

Recently, a sketch benchmark for 250-objects was released and analyzed in a similar recognition pipeline [Eitz et al., 2012b]. The authors proposed a gradient-based fast histogram feature, similar to the SIFT feature, based on fast Fourier transform. It is reported that soft quantization and SVM outperforms hard quantization and k-nearest neighbor approach. As with traditional image recognition tasks, gradient-based shape descriptors work fine for shape recognition tasks. Another study about symbol classification examines the role of local and global shape descriptors [Battiato et al., 2015], where local SC features are clustered to obtain a BoW representation for aligning the shapes, and the Circular Blurred Shape Model (CBSM) [Escalera et al., 2011] is utilized to describe the shapes globally. The CBSM descriptor is defined via correlograms on radial sectors, and rings. The distances of each sampled point on the contour to the center of these radial regions are computed and normalized. The inverse of these distances compose the correlogram distribution. In [Battiato et al., 2015], this approach is applied on the 70-class MPEG-7 silhouette dataset, 17-class symbol dataset [Escalera et al., 2011], and the aforementioned sketch dataset. Their approach, which includes local aligning of the shapes before describing them globally, is reported as superior compared to the approach in [Eitz et al., 2012b], which does not include an aligning step.

Data-driven representations. Recently, an important trend in machine learning argues that features should be learned from the data rather than designed by hand [Bengio et al., 2013]. This has led to a development of methods that can leverage and enforce sparsity in the representation. The sparse autoencoder (SA), which is an unsupervised neural network, is one such approach. This includes several variants such as denoising autoencoder, contractive autoencoder, and their regularized stacked versions [Bengio et al., 2013]. There have been several promising applications of convolutional and stacked autoencoders to object recognition [Ranzato et al., 2007; Xie et al., 2015], 3D object retrieval [Leng et al., 2015], handwritten Chinese character recognition [Wang et al., 2014], and multiple organ detection [Shin et al., 2013]. A thorough study about single-layer sparse autoencoders was conducted in which SA is used to extract the representation of local patches in an image on the most often used natural image benchmarks [Coates et al., 2011]. This work suggests to use dense sampling, small stride (the distances between sampled patches), and small input patch size (so that the number of features can increase) for performance improvement. In [Chen et al., 2015], the authors discuss convolutional SA-based features for image matching task and compare them with SIFT features. The main finding is that even though SIFT performs better for the given configurations and parameter settings, SA-based features have potential to compete, and that with a larger number of hidden units, SA features give better performance. On the other hand, assessing the performance of knowledge-driven descriptors compared to data-driven representations for shape data is still an open issue.

In this chapter, we propose to investigate the use of SA methods for the recognition of complex shape images. This differs from the studies on MNIST hand-written digits or Chinese characters, because in those studies, shapes are simple and small enough to be handled as a whole rather than patches. In our case, SA features are generated on patches similar to [Coates et al., 2011] and [Chen et al., 2015]. However, their focus is on natural images and ours is on complex binary shape data that have neither color nor texture.

3.3 Methodology

We focus on two representations: the previously proposed HOOSC descriptor; and a convolutional representation learned by a single-hidden-layer sparse autoencoder. Our aim is to study how a fully automatic learned representation competes with hand-designed features for shapes. This is done in a traditional bag-of-words (BoW) recognition scheme. In this section, we first describe our recognition scheme, and then explain the HOOSC descriptor and the SA representation.

3.3.1 Shape Classification Method

The classification method is illustrated in Fig. 6.1. The notation is summarized in Table 3.1. Given an input shape image I to be classified, the following five steps are applied.

1. Sample a set of positions $P = \{p_1, ..., p_i, ..., p_{N_P}\}$ over the image I, where N_P is the number of samples.
2. At each position p_i, extract a feature vector f_{p_i} as described in either Section 3.3.2 or Section 3.3.3, and compose the overall feature vector $F_I = [f_{p_1}, ..., f_{p_i}, ..., f_{p_{N_P}}]$.
3. Quantize each descriptor f using a dictionary D containing N_D elements to obtain the quantized indices $Q_I = [q_{p_1}, ..., q_{p_i}, ..., q_{p_{N_P}}]$, where each $q_p \in \{1, ..., N_D\}$.
4. Compute the histogram b_I from Q_I, where $b_I(q)$ contains the number of times q appears in Q_I.
5. Classify b_I into one of the N_c shape classes.

Note that the feature vector f_p around each point p is computed from a local circular patch x_p centered at p, and whose size is defined by the scale sc, defined as a ratio. That is, $sc = 1/1$ means that the patch size is equal to the input image size (defined by the longest edge of the

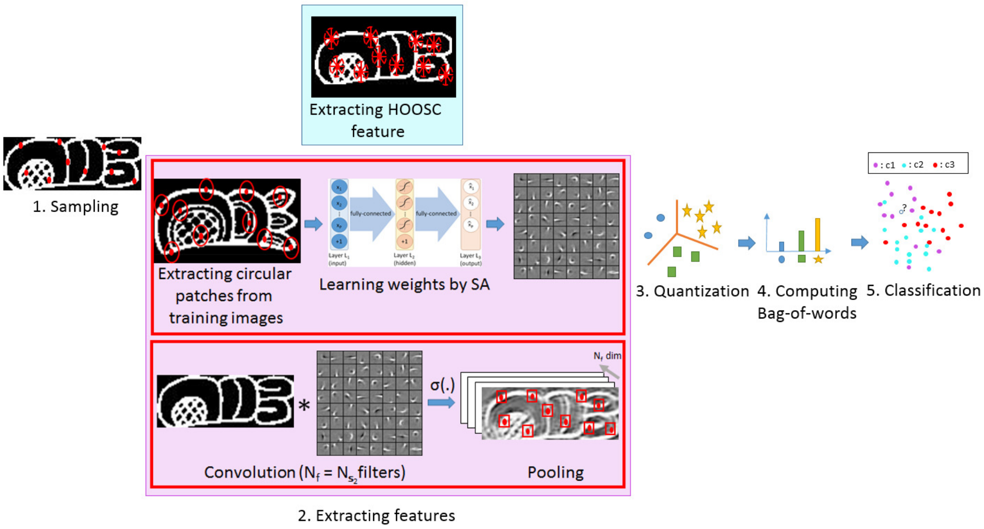


Figure 3.1 – Steps of the overall classification system. For feature computation, either the HOOSC (blue) or the SA (purple) methods can be used. The SA approach requires a learning phase.

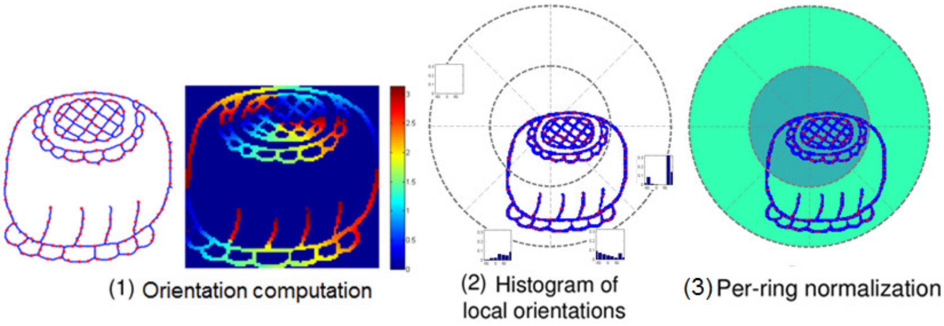

(1) Orientation computation
(2) Histogram of local orientations
(3) Per-ring normalization

Figure 3.2 – HOOSC computation at a sample position p of the shape, which includes: computation of 1) pixelwise orientations, 2) histogram of local orientations in each spatial bin, and 3) per-ring normalization of the histograms.

image), $sc = 1/2$ means that it has half the size, and so on. In addition, some of the steps require training. More precisely, in step (2), while the HOOSC computation follows from its definition (see Section 3.3.2), the SA representation requires learning (as described in Section 3.3.3). Similarly, before step (3), the dictionary D is learned by clustering a set of features f_p sampled from training images using the k-means algorithm. The learned dictionary D is used in step (3) to quantize each feature f in the overall feature vector F_I. Finally, in step (5), we classify the b_I representation of the image into a shape class using the k-nearest neighbor method.

Table 3.1 – Notations used in this chapter.

Notation	Explanation	Applicable to / Used for
I	Input image	HOOSC, SA
p_i	i^{th} position	HOOSC, SA
P	A set of positions	HOOSC, SA
F_I	Overall feature vector	HOOSC, SA
f_{p_i}	Feature extracted for p_i	HOOSC, SA
N_f	Number of dimensions of feature vector	HOOSC, SA
D	Dictionary	HOOSC, SA
N_D	Number of elements in the dictionary D	HOOSC, SA
Q_I	Quantized indices for all sampled positions in image I	HOOSC, SA
q_{p_i}	Quantized indices for p_i in image I	HOOSC, SA
N_P	Number of samples	HOOSC, SA
b_I	Histogram of bag-of-words for image I	HOOSC, SA
sc	Spatial context (spatial extent), see Section 3.3.1 for definition	HOOSC, SA
N_c	Number of classes	Classifier
k	Number of neighbors in k-NN classification	Classifier
N_a	Number of bins for discretizing oriantation angles	HOOSC
N_r	Number of rings	HOOSC
N_s	Number of spatial slices	HOOSC
${a_i}^{(l)}$	Activation of unit i in layer l	SA
L_i	The i^{th} layer	SA
N_l	Number of layers l	SA
N_{s_l}	Number of units in layer l	SA
$W^{(l)}$	Weights of layer l	SA
$b^{(l)}$	Bias of layer l	SA
$\sigma(.)$	The activation function (sigmoid)	SA
$h_{W,b}(x)$	Estimated target values for x with W and b parameters	SA
x_i	Value of i^{th} example (a pixel value of a sampled patch)	SA
N_{tr}	Number of training examples	SA
$KL(\rho\|\|\hat{\rho_j})$	KL divergence between sparsity and estimated sparsity	SA
ρ	Sparsity parameter (target mean activation of SA)	SA
β	Coefficient for sparsity cost term	SA
α	Weight regularization parameter (weight decay)	SA

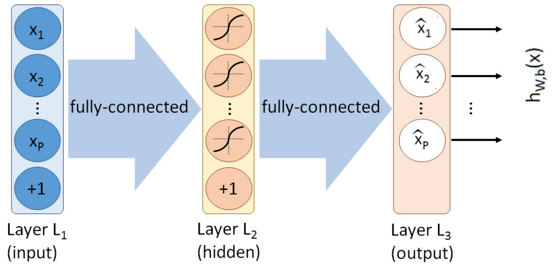


Figure 3.3 – Sparse autoencoder model. Layer 1 is the input layer (each blue input unit holds the value of one pixel of an input local patch in our application), Layer 2 is the hidden layer, and Layer 3 is the output layer where the reconstructed output activations (which should match the input vector size) are computed (drawing modified from [Ng, 2013]).

3.3.2 Handcrafted Features: Histogram of Orientation Shape Context

The Histogram of Orientation Shape Context (HOOSC) is a shape descriptor previously proposed for Maya glyphs [Roman-Rangel et al., 2011a]. The HOOSC feature is computed in two main steps as illustrated in Fig. 3.2.

First, the orientation of all image points is computed. This is done using the software by [Kovesi, 2015] ("featureorient" function), which extracts orientations from the local tensor of gradient directions, after image smoothing using a Gaussian filter. Note that this method differs from the original HOOSC definition [Roman-Rangel et al., 2011a], in which the orientations are computed using simple pairwise pixel differential operations, for which the shapes are thinned to their skeletons by morphological operations. Our method improves the local orientations around thick strokes, which avoids the unwanted branches that are usually brought in by the thinning process (the pivots on these branches end up having different orientations even if they are on the same thick contour area and have similar orientations).

Secondly, a histogram of local orientations is computed using N_a angle bins within each spatial bin of a circular grid centered at p_i. The HOOSC descriptor is obtained by concatenating all the histograms, and by applying a L_1 normalization for each of the two rings. Note that HOOSC is characterized by several parameters: the distance sc of the spatial context defining the extent of the spatial partition; the number of rings N_r; and the number of slices in a ring N_s. In our experiment, we use $N_a = 8$, $N_r = 2$, $N_s = 8$, which resulted in a HOOSC descriptor of $N_f = 128$ dimensions.

3.3.3 Feature Learning: Sparse Autoencoder

The SA model and its application to shape classification are described below.

Sparse Autoencoders

We introduce the SA model following the notation and formulation provided in [Ng, 2013]. An autoencoder is a feed-forward unsupervised neural network. Neural networks are composed of three types of layers: an input layer, hidden layers, and an output layer. Units are the building blocks of the neural network architecture; in each layer, they compute an output value from outputs of the previous layer, and are connected to all units in the next layer. Let ${a_i}^{(l)}$ be the output value of unit i in layer l.

For a given input x, the single hidden layer autoencoder is illustrated in Fig. 3.3 and defined as follows. The first layer L_1 is the input layer. The units in L_1 give the input values as their output. These input values are low-level image features, such as pixel values in general. Thus we have:

$$a^{(1)} = x. \tag{3.1}$$

The second layer L_2 takes the output values of $a^{(1)}$, applies a linear transformation (multiplication by a weight matrix $W^{(1)}$ and addition of a bias vector $b^{(1)}$), and then passes these values through a linear or nonlinear function, such as the sigmoid function $\sigma(z) = \frac{1}{1+exp(-z)}$, leading to:

$$a^{(2)} = \sigma(W^{(1)}a^{(1)} + b^{(1)}). \tag{3.2}$$

In general, each element of the weight matrix $W_{ji}^{(l)}$ is the parameter that connects unit i in layer l to unit j in layer $l+1$. Finally, the third layer L_3 is the output layer and the units in L_3 outputs the reconstructed input values $h_{W,b}(x)$ by following the same linear transformation (but not applying the sigmoid function) as in the case of the second layer units. We have thus:

$$h_{W,b}(x) = W^{(2)}a^{(2)} + b^{(2)} \tag{3.3}$$

When the weights are tied, the transpose of the weight matrix in L_2 is used for the weights in L_3, $W^{(2)} = W^{(1)T}$. We followed this approach, and hence the autoencoder is parameterized by $W = W^{(2)} = W^{(1)T}$, and $b = (b^{(1)}, b^{(2)})$. The weight matrix is learned during optimization by back-propagation.

An autoencoder differs from a traditional neural network, which outputs a class label. The autoencoder, instead, approximates its input x (layer L_1 in Fig. 3.3) by reconstructing the output $\hat{x}$ (layer L_3 in Fig. 3.3). The aim of this set up is to capture structure of the data in the hidden layers of the autoencoder (layer L_2 in Fig. 3.3). In another perspective, SA can be stated as composed of two parts, namely encoder and decoder. The coder maps the input data to the “codes” in the hidden units, i.e., the representation, and the decoder maps back these codes to the output data by minimizing the reconstruction error.

Sparse autoencoders (SA) are regularized autoencoders with an additional sparsity penalty term in the cost function. Let N_{tr} be the number of training examples x_i. Then the SA cost function can be expressed as follows:

$$J_{sparse}(W,b) = J(W,b) + \beta \sum_{j=1}^{N_{s_2}} KL(\rho \parallel \hat{\rho_j}), \tag{3.4}$$

$$J(W,b) = \left[\frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \left(\frac{1}{2} \left\| h_{W,b}(x_i) - y_i \right\|^2 \right) \right] + \frac{\alpha}{2} \sum_{i=1}^{N_{s_2}} \sum_{j=1}^{N_{s_3}} \left(W_{ji} \right)^2, \tag{3.5}$$

where $J(W,b)$ stands for the energy term for neural networks parameterized by the weight matrix W and biases $b = (b^{(1)}, b^{(2)})$. It computes the average reconstruction error and includes a regularization term on the weight matrices. In Eq. 3.4, $KL(\rho \parallel \hat{\rho_j})$ denotes the Kullback-Leibler divergence [Kullback and Leibler, 1951] between the a priori distribution of the activation unit $a_j^{(2)}$ modeled by a Bernoulli distribution with mean ρ, and the empirical distribution whose parameter $\hat{\rho_j}$ is computed from the training data ($\hat{\rho_j} = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \left[a_j^{(2)}(x_i) \right]$). The KL divergence is given by $KL(\rho \parallel \hat{\rho_j}) = \rho \frac{\log \rho}{\log \hat{\rho_j}} + (1-\rho) \frac{\log 1-\rho}{\log 1-\hat{\rho_j}}$.

The divergence measures how distant two distributions are. Thus, by choosing empirically the sparsity parameter ρ to be small and close to zero, we can enforce the codes $a_j^{(2)}(x)$ to be inactive (zero or close to zero) most of the time. In Eq. 3.4 and 3.5, several parameters control the importance of the different cost terms: β controls how strong the sparsity penalty term is, while α controls the regularization level.

Training. We utilized the libORF machine learning library [Firat, 2015] written in MATLAB. A sigmoid function is utilized as nonlinear hidden unit activation function in the model. The model is learned by backpropagating the parameters through the network. For optimizing the parameters, the mini-batch Stochastic Gradient Descent (SGD) method is utilized. The batch size is empirically chosen as 100, and the optimization is performed for 200 epochs. In order to get away from local minima and saddle points and for faster convergence, momentum and adaptive learning rate methods (namely, Adadelta [Zeiler, 2012]) are used. The momentum parameter controls how much the new gradient is affected by the direction and the magnitude of the gradient in the previous step in the optimization, and it is empirically chosen as 0.9. We note that while the limited Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm might

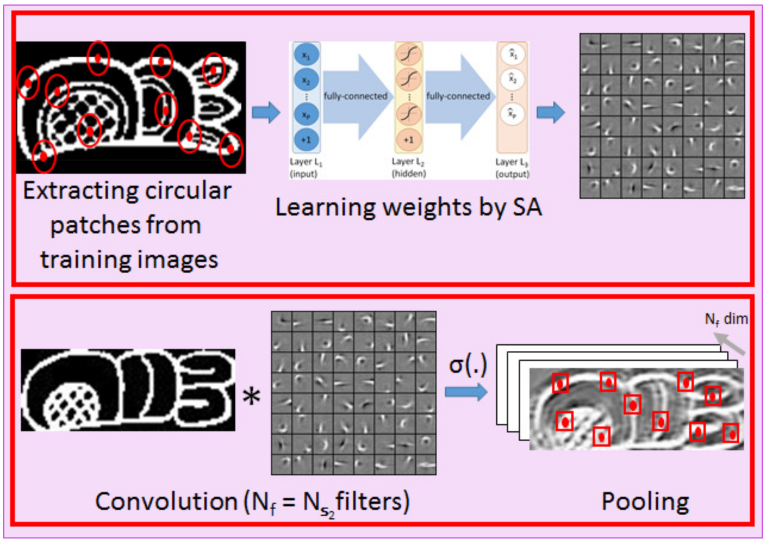


Figure 3.4 – Applying SA for shape classification. Top box: learning the autoencoder parameters. Bottom box: applying the autoencoder filters to a new shape image.

provide better-converged weights (smaller test reconstruction error) than SGD [Ngiam et al., 2011], it is computationally demanding (in terms of memory usage), so it was discarded as a choice.

Application to Shape Classification

Applying the autoencoder model to the shape classification task requires two steps: how we learn the encoder parameters (learning step) and how we compute the features for new shape images (feature extraction step). These steps are illustrated in Fig. 3.4. Applying the SA model requires its input area to be the same as HOOSC's. As we use the same input patches, the SA representation can then be fairly compared to HOOSC.

Learning. To learn the weights W, circular patches centered around sampled positions p and spanning the defined spatial context are cropped from the images. These patches are used as input to the single-layer SA and the weights are obtained as described in the previous subsection. To understand the representation learned by SA, the learned weights are visualized in Fig. 3.4 (top, rightmost). Each square j of the 8x8 matrix displays the normalized weight values $\hat{W}_{ji}$ connecting the input image x to the activation function $a_j^{(2)}$ of the hidden unit j defined as:

$$\hat{W}_{ji} = \frac{W_{ji}^{(1)}}{\sqrt{\sum_{i=1}^{N_p}(W_{ji}^{(1)})^2}}. \tag{3.6}$$

Note that the activation unit $a_j^{(2)}$ is maximal when the input patch x indeed corresponds to $\hat{W_{ji}}$. We can thus interpret the visualization of each small square as the type of structure that the corresponding hidden unit will be responsive to. Accordingly, we treat each of these squares as spatial filters that can be applied to an input image. The representation is not only the filter, but involves going through the sigmoid function $\sigma(.)$ to get the activation $a_j^{(2)}$.

Feature extraction. Given a shape image, we need to compute the SA representation at a set of positions p_i of the image (see Section 3.3.1). Rather than extracting a patch x_{p_i} around each point and computing its representation f_{p_i} (which is provided by the activations vector $(a_1^{(2)},...,a_{N_{s_2}}^{(2)})$), we proceed as follows. To compute the feature map f_{m_j} corresponding to activation unit j, we convolve the image with the filter W_j, and pass the output through the sigmoid function $\sigma(.)$. The second step is to pool (i.e, to aggregate) the feature map responses of each filter. In pooling, the responses within a small neighborhood around the sampled positions p_i are converted to feature vectors f_{p_i} by taking their sum, max, or just taking the response at the sample positions.

3.4 Datasets and Experimental Protocol

While our main interest is in analyzing Maya glyphs, we also want to understand the generalization ability of the representations to other shape categories. With this in mind, we have assessed the HOOSC and SA representations on two datasets, namely the Maya monument glyph dataset and the sketch dataset. This section describes the datasets and the experimental settings.

3.4.1 Datasets

The data in our experiments are binary shapes, more specifically contours. Due to the lack of color and textural information, this data source has other challenges compared to natural images. The main issues are the visual variations due to missing or extra parts across samples, different writing styles across regions or time (see Fig. 2.3), or different viewpoints (in the case of sketch data).

Maya Glyphs

The Maya glyph dataset used in this chapter was introduced in the the Section 2.4 of the previous chapter.

All the images are resized to be enclosed in a 128x128 pixel area, i.e. the largest glyph side is set to 128 pixels. To characterize the shapes, we measured the stroke thickness using morphological operators. For this dataset, the mean average stroke thickness is 2.2 pixels, and the mean of the maximum stroke thickness in each image is 5 pixels.

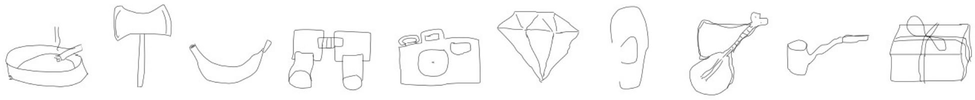


Figure 3.5 – Representative images of the randomly selected 10 classes from sketch dataset [Eitz et al., 2012b].

For Maya glyph classification, it is challenging to obtain a large dataset with enough samples for classification task due to the limited amount of sources and the amount of expert work needed to produce data. In this chapter, we focused on the previously collected dataset from [Roman-Rangel et al., 2011a] as a source of Maya glyph dataset.

Sketches

Eitz et. al. provide a crowdsourced human sketch dataset over 250 object classes [Eitz et al., 2012b]. Each class is composed of 80 sketches. For our experiments and to compare with the Maya glyph case, we use two versions of this dataset: a 10-class subset, and the full dataset. For the first case, we randomly picked the 10 classes. These classes are: ashtray, axe, banana, binoculars, camera, diamond, ear, guitar, pipe (for smoking), and present. For this dataset, the mean average stroke thickness is 1.2 pixels and mean maximum stroke thickness is 2 pixels. In other words, these sketches are roughly twice as thin as the Maya glyphs, and due to the way they were produced (digital pen of fixed thickness), contain almost no thickness variations (see Fig. 3.5). The original samples are provided as square images (1111x1111) in which the shapes are in the center and padded by 120 pixels (on average) on each side. We rescale the images to 128x128 size in our experiments.

3.4.2 Classifier

Given the BoW representation of each image, we used a k-nearest neighbor (k-NN) classifier with L_1 distance metric. This is illustrated in the fifth step of Fig. 6.1, where samples from various classes are shown in different colors, and the label of the test sample (shown in question mark) needs to be inferred from the labels of its k nearest neighbors. In the case of a tie between two or more classes, any of the tied classes is randomly assigned. The motivation for choosing k-NN as classifier is two-fold: the risk of overfitting problem due to small amount of data, and the emphasis of the comparison being on the visual representations (HOOSC vs. SA) and not necessarily on the machine learning schemes.

3.4.3 Performance Measure

The classification results are evaluated using average accuracy across data splits. The classification accuracy is computed as the ratio of the number of correctly classified samples (both positive and negative samples) over the total number of test samples.

3.4.4 Evaluation Protocol

For the glyph dataset, we randomly selected 25 image samples from each category for training, and the rest (413 samples) are used for testing. We repeated this process three times, and the average accuracy of the three data splits is reported. The SA representation has been trained on patches extracted from a total of 630 images, coming from the training samples of the 10 classes (250 images) as well as the unexploited images of the other 14 classes of the dataset (380 images), since training such a network requires as many samples as possible.

For the 10-class sketch dataset, we randomly selected 50 image samples from each class for training, and used the remaining 30 images for testing per class. The SA representation for sketch experiments are trained using 50 patches from each of the 50 image training samples of each class of the full set, thus resulting in 250x50x50 = 625000 patch samples.

For the 250-class experiments, we followed the experimental procedure described in [Eitz et al., 2012b] (3 times run of 3-fold experiments) with 48 training samples per class, and reported the average accuracy.

3.4.5 Parameter Setting

In this section, we describe the parameter choices selected to conduct the classification experiments.

Spatial Context

One key parameter to study is the amount of spatial support that is used to build a descriptor (HOOSC or SA) around each point sample (termed as "spatial context"). In our experiments, the spatial context is defined as a fraction $sc \in \{1/1, 1/2, 1/4, 1/8\}$ of the longest image edge in both datasets. Since all images are scaled to 128 pixels while their aspect ratio is kept the same during preprocessing, the circular patches have fixed diameters (128, 64, 32, and 16, respectively) in each spatial context level. Fig. 3.6 illustrates the regions covered by these spatial context levels for the two different examples. Fig. 3.7 shows 100 example patches for each spatial context level and for both datasets.

Sampling Strategy

To build the BoW representation of an image, we need to sample the image positions where the local descriptors (the HOOSC or the SA representation) are to be computed and quantized. Two sampling strategies are evaluated. In the first one, following the previous work on shape analysis [Belongie et al., 2000], the points (pivots) are sampled randomly along the contour. We work with 300 pivots in our experiments. Furthermore, for fairness in the experiments comparing the HOOSC and the SA representations, the same exact pivots have been used. As

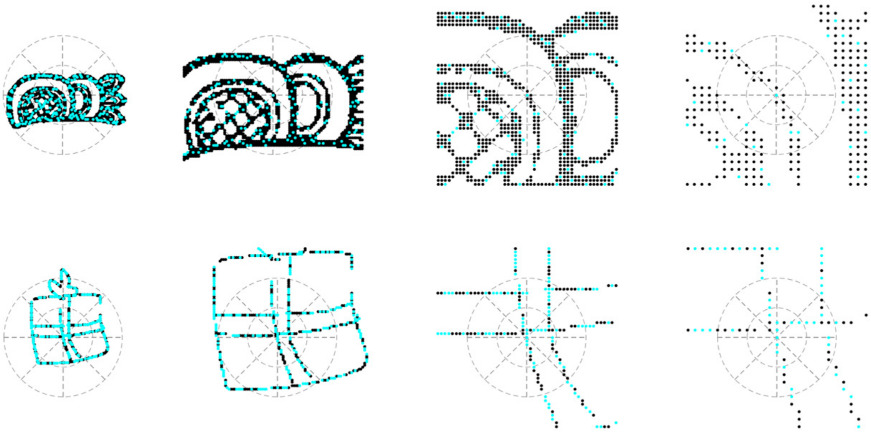


Figure 3.6 – Region used to compute the patch descriptor (HOOSC or SA) around a sample point (cyan) for the four different spatial context levels (from left to right, *sc* of 1/1, 1/2, 1/4, and 1/8) for a glyph and a sketch sample.

Table 3.2 – Receptive field (patch) sizes while learning weights in SA.

Spatial context (sc)	sc = 1/1	sc = 1/2	sc = 1/4	sc = 1/8
Patch size in image	128	64	32	16
Actual patch size as input to SA (after rescaling)	32	32	32	16

the second strategy, following [Coates et al., 2011] and [Eitz et al., 2012b], dense sampling is applied. For dense case, the stride is set to 4 pixels, and 10-pixel offset is kept from the edges. In total, 28x28=784 regularly sampled pivots are used.

HOOSC Parameters

We follow the setting utilized in [Roman-Rangel et al., 2013], which was shown to perform best in the large majority of cases. The descriptor is built using 2 rings, 8 spatial bins, and 8 orientation bins.

Sparse Autoencoder Parameters

Training the SA representation using input patches of large sizes quickly leads to a large number of parameters to be learned. Roughly speaking, using circular patches of diameter N pixels, and N_D hidden units results in $O(N^2 N_D)$. For $N = 32$ and $N_D = 64$, the number of parameters is 65536 as noted in Table 3.5. Thus, in practice, while training SA, we keep the patch size to be the same for different spatial context levels (except $sc = 1/8$) due to the increasing model complexity and the data shortage limitation. When dealing with patches of spatial context $sc = 1/1, 1/2$ levels, we rescaled their content to the actual patch size of sc =

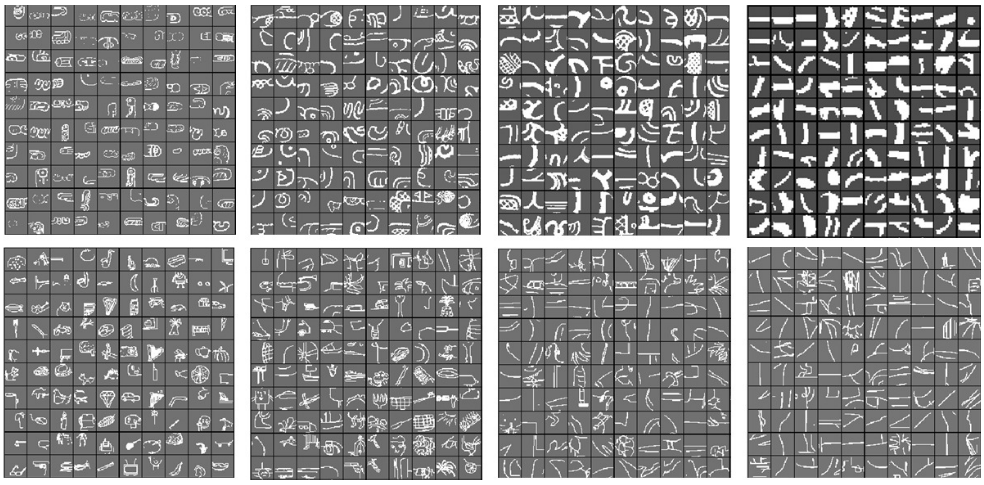
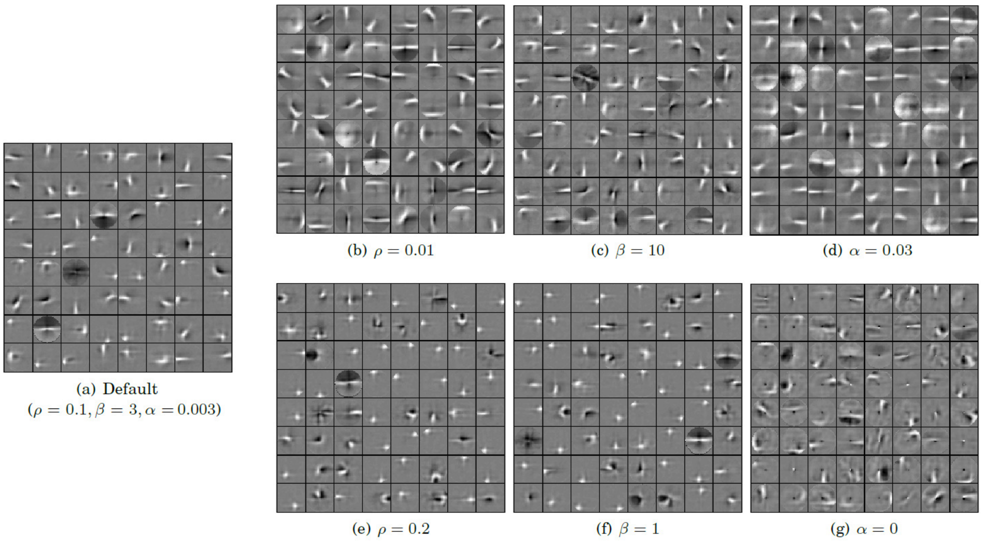


Figure 3.7 – Examples of circular glyph patches (top row) and sketch patches (bottom row) given as training data to SA when using a spatial context *sc* of $1/1, 1/2, 1/4, or 1/8$ (from left to right).



(a) Default ($\rho = 0.1, \beta = 3, \alpha = 0.003$)

(b) $\rho = 0.01$

(c) $\beta = 10$

(d) $\alpha = 0.03$

(e) $\rho = 0.2$

(f) $\beta = 1$

(g) $\alpha = 0$

Figure 3.8 – Impact of the SA parameters on the learned weights (filters) when using 64 hidden units and the glyph dataset. On the left, the weights learned with the default parameters are shown. On the right, the impact of each single parameter keeping the others fixed is illustrated in each column (the rest of the parameters have default values).

1/4. For both datasets (Glyph and Sketch), the input to SA for $sc = 1/1, 1/2, 1/4$, is provided as patches of 32 pixels of diameter; and for $sc = 1/8$, the patch size is 16 pixels.

To train a SA representation, we need to set three parameters introduced in Section 3.3.3, namely the sparsity parameter ρ, the sparsity cost term coefficient β, and the weight regularization parameter α. To analyze the impact of these parameters on training, we set the spatial

Table 3.3 – Selected parameters for learning weights in SA.

Parameters	Sparsity parameter (ρ)	Sparsity cost term coefficient (β)	Weight regularization parameter (α)
Value	0.1	3	0.003

Table 3.4 – Correlation values of the weights learned with different parameters (shown in Fig. 3.8).

Parameter values	L1 distance of eigenvalues of the correlation matrix to identity
(a) Default ($\rho = 0.1, \beta = 3, \alpha = 0.003$)	45.4
(b) ($\rho = 0.01, \beta = 3, \alpha = 0.003$)	59.8
(c) ($\rho = 0.1, \beta = 10, \alpha = 0.003$)	52.2
(d) ($\rho = 0.1, \beta = 3, \alpha = 0.03$)	87.1
(e) ($\rho = 0.2, \beta = 3, \alpha = 0.003$)	39.9
(f) ($\rho = 0.1, \beta = 1, \alpha = 0.003$)	41.2
(g) ($\rho = 0.1, \beta = 3, \alpha = 0$)	39.2

context sc to 1/4 and consider 64 hidden units for the glyph data. Fig. 3.8 shows the weights learned with different parameter values.

On the left side of Fig. 3.8, the chosen weights are illustrated. Then, the weights learned with two other values of the parameters are shown in the following three columns. In each column, only one parameter is changed (stated below each subfigure) and the other parameters are set to their default values (see Table 3.3). Looking at the filters, we observe a similar effect when a larger sparsity parameter ρ or a smaller sparsity cost term coefficient β are utilized. Since this corresponds to putting less weight on the sparsity term in the objective function, we obtain more localized filters. In other words, the reconstruction of the input data with these filters would be as if putting pixelwise responses from each hidden unit together. On the other hand, as ρ is smaller and β is larger, each filter is implicitly required to account for more spatial scope and therefore becomes less localized, in other words, more “blurry”. As the regularization parameter α gets larger, we observe less variation between the weight values (less sparse weight per filter).

To derive a good set of parameters, we analyzed the correlations of the weights. We computed the correlation matrix of the filters (the inputs that would give the highest activation from the learned weights of each hidden unit) for each parameter configuration. Our aim is to select the configuration that results in a correlation matrix whose eigenvalues are small [Bengio and Bergstra, 2009], i.e., decorrelated weights. The L1 distance of the eigenvalues of the correlation matrix of the weights to the identity matrix is shown as a quantitative measure in Table 3.4. In this table, we observe that this measure gives smaller values for more localized filters (such as cases (e) and (f) in Fig. 3.8) compared to other configurations. In view of this, we set the parameter configuration such that this measure gives a small value and the filters are not

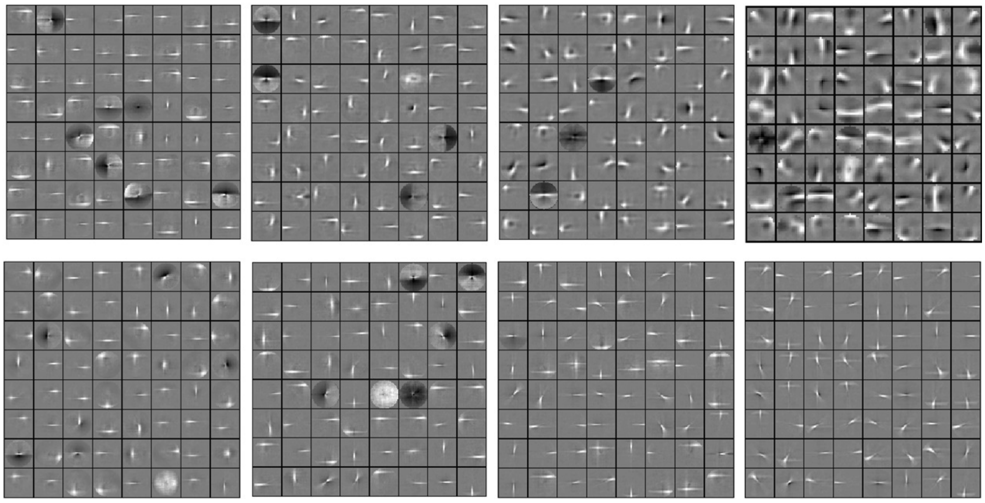


Figure 3.9 – Tied weights learned by SA for the patches with spatial context *sc* of 1/1, 1/2, 1/4, and 1/8 (from left to right). The top row is for glyph data and the last row is for sketch data.

Table 3.5 – Number of parameters while learning weights in SA.

Dataset	Number of training samples	Patch Size	Number of hidden units	Number of parameters (Tied)
Glyph	630*300=189000	32	64	32x32x64=65536
Glyph	630*300=189000	32	256	32x32x256=262144
Sketch	250*50*50=625000	32	64	32x32x64=65536
Sketch	250*50*50=625000	32	256	32x32x256=262144

too localized. Table 3.3 shows the chosen parameters. These parameters were used for all experiments with SA on both datasets.

3.5 Results and Discussion

This section describes and discusses the performance according to variations of model parameters and evaluation frameworks. We first report results on the 10-class experiments for the Maya and sketch dataset. We then consider the extension to the 250 classes of the sketch data.

3.5.1 10-class Experiments

Overall Results

Fig. 3.10 shows the 10-class classification results for the 4 different spatial context levels of the representations for both glyph (left) and sketch (right) datasets. Fig. 3.10 also illustrates the comparative analysis of two different representations, namely HOOSC and SA with different number of hidden units (SA-64 and SA-256 with average pooling). Unless stated otherwise, for

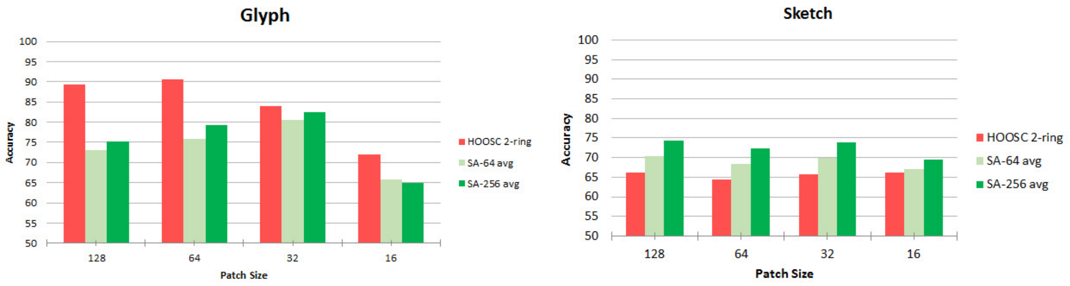


Figure 3.10 – 10-class classification results of the SA (64 or 256 hidden units), and the HOOSC descriptors with four spatial context levels for glyph (left) and sketch datasets (right). The number of words in quantization is 500, and $k = 5$ for k-NN classification.

building the dictionary D via k-means clustering, the number of elements in the dictionary (vocabulary size) N_D was set to 500, and for k-NN classification, 5 nearest neighbors ($k = 5$) were used. The experiments are repeated three times with random splits of the samples and the average accuracy is reported.

HOOSC vs SA Representations. The HOOSC descriptor and the SA representations have different trends in the classification accuracies for both datasets. In Fig. 3.10 (left), we observe that the HOOSC descriptor outperforms the SA representations in the glyph case, but the SA representation performs better than the HOOSC descriptor for the sketch data. This might be due to the HOOSC descriptor being a knowledge-driven representation. Since the HOOSC representation was originally proposed and designed for the glyph data, the parameters of the HOOSC descriptor are chosen optimally for glyph samples, even though those parameters might not be the optimal choice for the sketch data.

Glyph vs Sketch Data. Fig. 3.9 shows the filters learned with the patches collected on four spatial context levels for glyph (top row) and sketch (bottom) datasets. We observe curvy patterns on some filters, especially at the last two smaller spatial context levels in the glyph case. Other than that, all filters are quite similar (corresponding to lines in various orientations), which suggests that these two datasets are not so different from each other with respect to local shape and that the local shapes do not exhibit much data-specific correlation or repetitions.

The main difference between these two datasets is the stroke thickness as observed from the filters learned from patches with smallest spatial context ($sc = 1/8$). Glyph filters learned from the smallest patches are thicker than sketch filters learned from the same spatial context level. Indeed, with glyph data, at smaller spatial context levels ($sc = 1/4$ and $sc = 1/8$), more circular and curvy activation functions are learned for SA. These curvy patterns can be observed for spatial context level $sc = 1/4$ in the sample patches in Fig. 3.7, although for the glyph with $sc = 1/8$, there are still curvy patterns, they are visually more similar to sketch patches with $sc = 1/8$ except for the stroke size. This kind of curvy details is not captured at the larger levels ($sc = 1/1$ and $sc = 1/2$). This is probably due to loss of details (like small circles) during rescaling of the patches. We also hypothesize that as the region where patches are collected gets larger, the continuity and angle of the contour become more prominent features rather than small details. Besides, the repeatability of small, curvy patterns at the same location in the training samples when using larger spatial context might be lower than in the smaller context cases. On the other hand, the weights learned on the sketch data do not contain any such small circular strokes, but mostly horizontal/vertical line strokes or intersection of these. We hypothesize this is due to the large variation in the sketching styles.

Even though glyph data is complex, it follows the Maya language rules. For instance, small circular patterns are useful parts to categorize some glyph classes. Based on the location or configuration of these parts with respect to others, the category of the glyph may change. For instance, the main visual difference of */ya/* (T126) and */ji/* (T136) are the small circles between half-moon shapes (see Fig. 2.7). Another example can be subtle differences between */li/* (T82), */ki/* (T102), and */ko/* (T110). The distinct points of */li/* from */ki/* are the small circles which are in the middle of the contour but not connected to any vertical "band" and the inner lines connected to the outer contour rather than the "band" as in */ki/*. Similarly, */li/* has an eliptic element which is connected to the outer contour. This element is also visible in */ko/*. The only difference between */li/* (T24) and */ka/* (T25) is the number of inner slanted lines, and for */ni/* (T116) and */wi/* (T117), the only distinctive hint is the direction of the small part. There are also compositionally similar glyph categories, even though visually they may seem not so close, as in the case of */ti/* (T59) and */tu/* (T92), which have three main parts. This means that there are shared, small patterns encoded in the glyph data. However, in the sketch case, shapes are by construction over-simplified or contain very few detailed samples. Furthermore, the shared patterns between sketch classes are not as obvious as in the glyph case. Therefore, only the lines with different orientations and their intersections are learned as common patterns by the SA architecture.

One point to recall is that the number of training samples are different in the two datasets. In the glyph dataset, there are 25 training samples, while in the sketch dataset we use 50 samples (see Table 3.5). We have also a larger quantity of data to train SA for the sketch case and less variability in stroke thickness. This suggests that the higher performance of the SA representation compared to the HOOSC descriptor on sketch data might be related to the larger amount of training data, which is also noted in [Coates et al., 2011; Eitz et al., 2012b]. Even though both algorithms perform worse on the sketch data as compared to the glyph case, performance of the HOOSC descriptor degrades much more as compared to the SA representation. Therefore, we hypothesize that the SA representation gains some performance by learning from a larger training set.

Impact of Spatial Context. The spatial context, i.e., the patch size where the representation is extracted, affects the HOOSC and SA representations differently in the two datasets (see Fig. 3.10). In the glyph case, as the patches get smaller, the performance of the HOOSC descriptor degrades, and it marginally increases in the sketch case. For the SA representations, we obtain the highest performance (78 % with SA-256 avg representation) with 32x32 patch size ($sc = 1/4$)

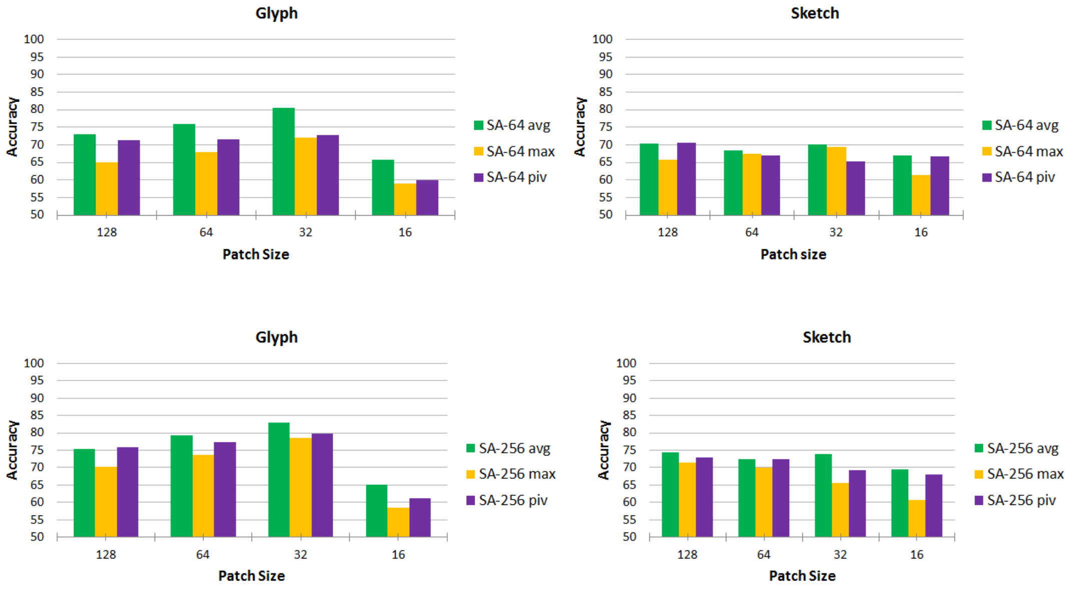


Figure 3.11 – 10-class classification results of the SA with 64 (top) or 256 hidden units (bottom) for three different pooling strategies and four spatial context levels, for glyph (left) and sketch datasets (right).

in the glyph case. However, for the sketch data, the trend is not so clear, as the performance values are close and minimal changes are observed across patch sizes. The learned SA filters (Fig. 3.9) play an important role on these performance values. Since the filters learned on sketch patches are similar, the performance values turn out to be close to each other. On the other hand, the filters learned on 32x32 ($sc = 1/4$) and 16x16 glyph patches ($sc = 1/8$) are the most characteristic filters (such as curvy patterns); we obtain the highest performance for the representations on 32x32 glyph patches, although the SA representations over 16x16 glyph patches are not competitive (as what the representation capture over the region becomes unspecific as can be seen in Fig. 3.6).

Analysis of the SA Representation

We now analyze the performance of the SA representation according to the number of hidden units during the learning step, and to the pooling strategy at the feature extraction step.

Impact of Number of Hidden Units. We can observe from Fig. 3.10 that a larger number of hidden units tend to increase the recognition accuracy. This is the case for both datasets and most spatial context levels sc, as shown in Fig. 3.10.

Impact of Pooling Strategy. The pooling operation is necessary to aggregate the convolutional filter responses and to obtain a reasonably-sized feature dimension. During pooling, the spatial information is also implicitly embedded, since the values are aggregated in 5x5

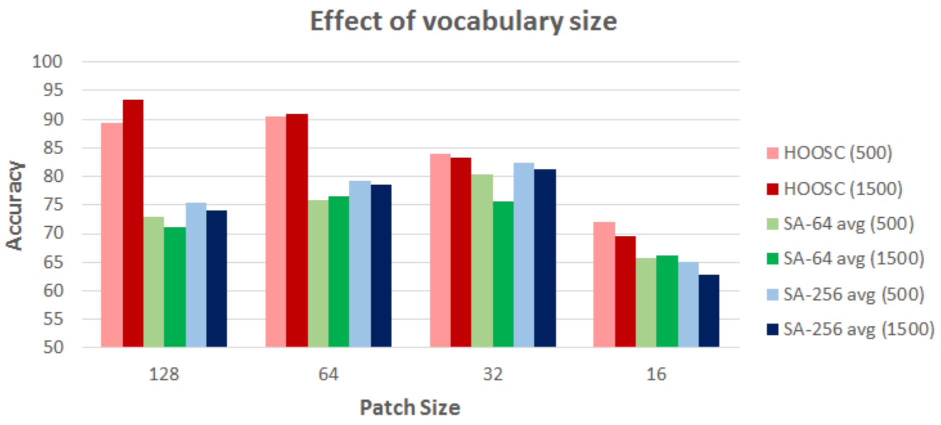


Figure 3.12 – 10-class classification results of the HOOSC and SA representations for 500-word and 1500-word dictionary size, with four spatial context levels, for the glyph dataset.

neighborhoods around the sampled pivots. We used three ways to get information from the circular region of interest: average pooling, max pooling, or taking only the response over the pivots. Results are shown in Fig. 3.11. We observe that in the great majority of the cases, average pooling outperforms or is on par with the other strategies. Furthermore, taking only the pivot response gives higher accuracy than max pooling except for the sketch SA-256 case. This is in contrast to recent work in the literature [Boureau et al., 2010], which empirically found that max pooling is robust to subtle changes in the pooled spatial neighborhood. We hypothesize that it is due to the nature of the data and our sampling strategy. Since the data consists of binary shape images sampled along contours, average pooled features might be more robust at representing and taking into account the contour variations around points compared to the max-pooled features.

Analysis of Bag-of-Words Model

Here, we analyze the bag-of-words model according to the dictionary size and the value of k in k-nearest neighbor classification.

Dictionary Size. In our experiments, the dictionary size does not follow a single trend. In Fig. 3.12, the performance values for glyph data are presented for $N_D = 500$ (light colors) and $N_D = 1500$ (dark colors) dictionary sizes. We observe 2 to 4 percent increase when using a larger vocabulary size as the patch sizes are large. HOOSC classification can reach a highest value of 93.46 % accuracy. However, for smaller patch sizes, a larger vocabulary size generally degrades or does not affect the results. Note that we also observed a similar trend in the sketch dataset.

Value of k in k-NN Classification. Fig. 3.13 shows the effect of the value of k in k-nearest neighbor classification for the HOOSC and SA-256 representations. To be brief, only the results

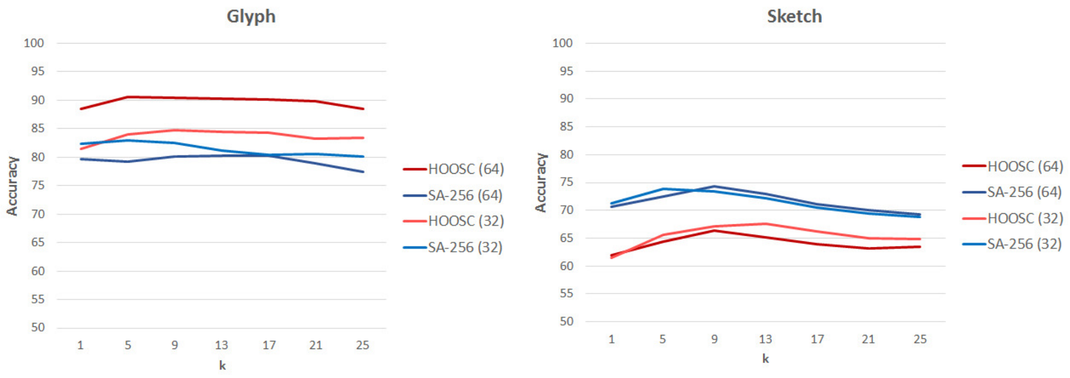


Figure 3.13 – 10-class classification results for the HOOSC and SA representations, for 64x64 ($sc = 1/2$) and 32x32 ($sc = 1/4$) patch sizes, with $k = 1,5,9,13,17,21,25$ during k-NN classification for glyph (left) and sketch (right) datasets.

of the representations defined on 32x32 and 64x64 patches are shown. Results are relatively stable overall with respect to k. We can however notice that for the representations defined on 32x32 glyph patches (light lines on left plot), the classification performance decreases as k increases. Similarly, the performance degrades for 64x64 sketch patches (darker lines on right plots) as k increases beyond a certain point.

3.5.2 Generalizing the Results: 250-class Experiments

We also study the generalization capability of our classification model over all 250 sketch classes. As stated earlier in the chapter, we designed a second set of experiments with a larger shape dataset in terms of numbers of classes and examples. Unfortunately, such dataset is not publicly available for Maya hieroglyphs, and so we use the sketch data as data source. For the future, we anticipate that large datasets of glyphs could be generated by experts. We followed the experimental procedure described in [Eitz et al., 2012b] (3-fold experiments) with 48 training samples per class, and reported the average accuracies. Besides, we compare our pivot sampling strategy with dense sampling.

How Do the Representations Generalize? The performance results of all the representations for 250 sketch classes are provided in Fig. 3.14, with vocabulary size equeal to 500, $k = 25$ during k-NN classification, and the same values in Table 3.3 used for SA parameters. As in the 10-class case, we observe that the SA representations clearly outperform the HOOSC representation (red) in the 250-class case (with 20.73 % for HOOSC and 29.36 % for SA). But there is also a significant drop in performance for all methods given the larger complexity of the task. The highest accuracy is of 29.36 %. As in the 10-class case, using more hidden units (256) leads to better results with most of the pooling strategies and patch sizes. Also similar to the 10-class case, average pooling performs better for the SA-64 representation. However, contrary to the 10-class results, the SA-256 pivot pooled representation is the most

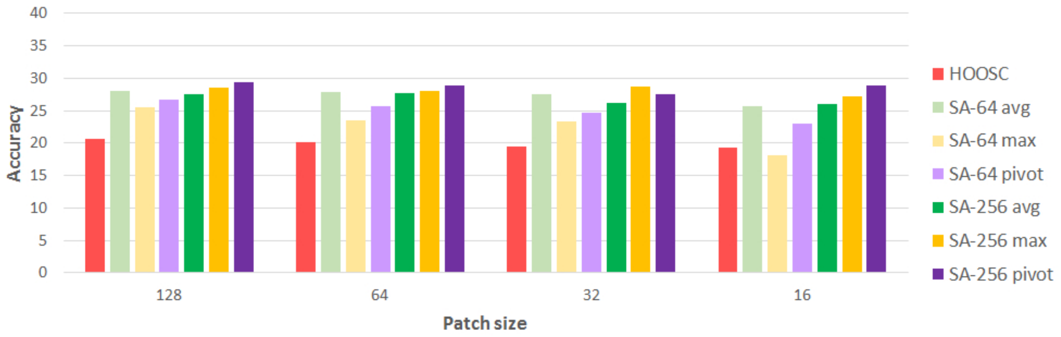


Figure 3.14 – Accuracy results of the HOOSC and SA representations with pivot-sampling and four spatial context levels for 250-class sketch classification.

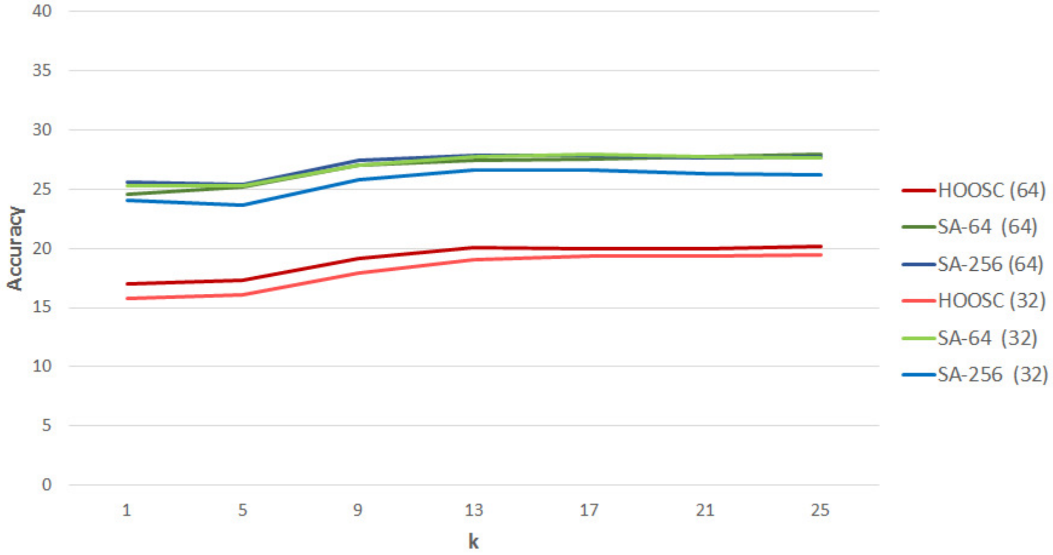


Figure 3.15 – 250-class sketch classification results of the HOOSC and SA representations, for 64x64 ($sc = 1/2$) and 32x32 ($sc = 1/4$) patch sizes with $k = 1, 5, 9, 13, 17, 21, 25$ during k-NN classification.

competitive of the SA-256 representations. The impact of patch size is not prominent in the 250-class case. Even though the HOOSC results degrade quite marginally as the patch size gets smaller, the SA results are close to each other in terms of the patch size they are defined on.

Value of k in k-NN Classification. Fig. 3.15 shows the corresponding plot of the right plot in Fig. 3.13 as the task is generalized to 250 sketch classes. This time, we see a clear trend with respect to the k value in k-NN classification. For both representations defined on both 32x32 and 64x64 patches, the performance increases with k value and comes to a stable state,

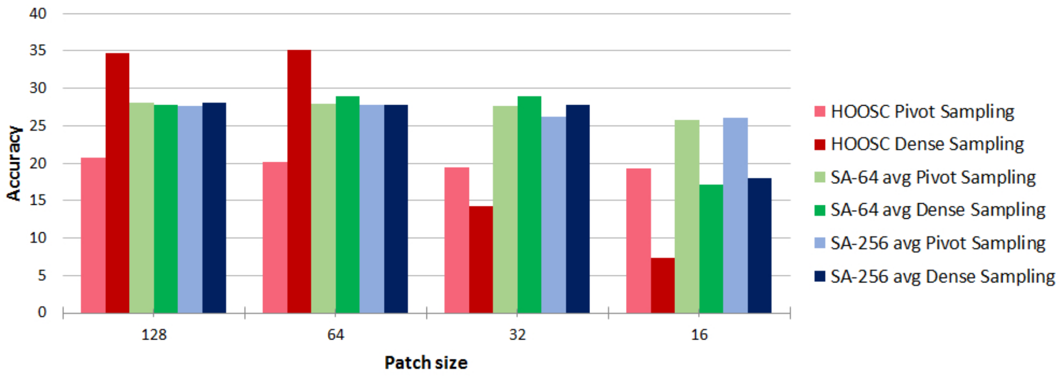


Figure 3.16 – 250-class HOOSC and SA results with densely-sampled and pivot-sampled patches (left) with four spatial context levels for sketch dataset.

reflecting that with more classes assessing the label with more neighbors is a better strategy.

Pivot vs Dense Sampling. The recent literature suggests that as the sampling of points around which descriptors are computed gets denser, the larger amount of produced descriptors results in a better model (here represented by the bag-of-words representation) of the overall image content [Coates et al., 2011]. We test the dense regular grid sampling (784 pivots) utilized in [Eitz et al., 2012b] for the HOOSC and SA representations as compared to random sampling on contours (300 pivots). The results are illustrated in Fig. 3.16. From this plot, we observe that the densely-sampled HOOSC descriptor defined over large regions is much better (about 15% absolute increase in accuracy) than the pivot-sampled HOOSC descriptor. However, the densely-sampled HOOSC results get worse than pivot-sampled HOOSC results as the patch size gets smaller. This is due to the nature of the shape data. Since shapes have large empty regions, as the patch size is smaller, the number of empty descriptors increases when using a regular grid sampling. This effect is also seen in the SA representation results with the smallest patch size (the dense sampling results are 7% lower than the pivot sampling results). Interestingly though, the SA representations are affected by this cause only at the smallest spatial context level (16x16). Furthermore, we observe that the SA results, regardless of the sampling, are quite similar (around 27-28% accuracy) for larger patch sizes.

Comparison with the state-of-the-art. We compare the studied representations with the local orientation-based descriptor in [Eitz et al., 2012b]. In their work, they use a square patch of around 90 pixels in size for 256x256 scaled images, which compares in terms of setting to our circular patches of radius 64. The accuracy obtained by the densely-sampled HOOSC over 64x64 patch regions is 35.13% which shows a marginal increase (around 0.63%) to [Eitz et al., 2012b] with their descriptor that uses hard quantization with 500 number of words.

3.6 Conclusion

In this chapter, we assessed a data-driven, non-linear filter bank representation against a hand-crafted, orientation-based histogram representation for Maya hieroglyphic shape data and sketch data. We investigated the impact of the different parameters of the representations and the most important aspects to consider while working with such image representations. The main conclusions were as follows:

- **Sparse Autoencoder modeling.** Appropriate parameters to learn the filters could be obtained by following principles in the literature, and few empirical choices. The learned filters reveal the data content, and are tuned to the inherent motifs in shape data. With respect to classification, both the 10- and 250-class experiments show that using a larger number of hidden units improved the performance in most of the settings on both datasets. In addition, average pooling tends to perform best.

- **Knowledge-driven vs. data-driven features.** Our work showed that the HOOSC descriptor performed better than the SA representation on the smaller dataset (glyph), for which the HOOSC descriptor was originally designed for. In contrast, when considering more data (10-class experiments on sketches) and more classes (250-classes), with the pivot sampling-based approach, the SA representation was able to surpass the HOOSC descriptor (29.36 % vs. 20.73 % accuracy on the 250-class experiments). This shows the importance (and the necessity) of the amount of available data for data-driven approaches.

- **Dense sampling.** With dense sampling, the HOOSC descriptor improved substantially when considering large enough spatial regions, reaching 35% accuracy on the 250-class case. On the other hand, the SA representation did not benefit from this factor, staying behind HOOSC's accuracy.

- **Comparison with the state-of-the-art.** The HOOSC performance is in par with that of [Eitz et al., 2012b] for dense sampling using big patches of sketch data. A probable reason why SA was not able to leverage the dense sampling on large patches is that increasing the patch size resulted in two issues. First, the low amount of stroke points that are present in many large patches obtained by dense sampling would make the learning of efficient representation difficult, as most contours would not tend to repeat in the training data. The second issue is that the number of model parameters grows quadratically with the patch size, making it infeasible to learn them from raw data. To avoid this problem, we downscaled the patches in the experiments, but at the cost of losing some useful information present in the original data. The main conclusion is that a single-layer SA is not sufficient to efficiently learn shape characteristics, and that deeper networks with more layers might allow a hierarchical and gradual learning of shape elements without the quadratic increase of parameters.

- **Use for cultural heritage.** We hope that our study, using both Maya hieroglyphics and

> generic hand-drawn shapes, can inform about the possible performance trends of the studied methods for other cultural heritage visual shape sources.

In this chapter, we observed that relatively large datasets are needed for data-driven approaches to be prominent against traditional visual representations. To investigate the data-driven approaches further in the Maya shapes case, we produced a relatively large individual Maya codical glyph dataset via crowdsourcing as described in detail in Chapter 4. Furthermore, while learning shape representations, we hypothesize that deeper architectures are needed in order to capture the regularities in the data better. In Chapter 5, we explore learning approaches based on deeper Convolutional Neural Networks in a supervised setting.

4 Crowdsourcing

To provide computational support to Maya scholars in visual analysis tasks, there is an evident need for large datasets, as concluded in Chapter 3. Large datasets would enable more capable shape representations and improve the performance of automatic analysis tools. However, considering the complexity of the Maya shapes, for the Maya experts to prepare a large dataset takes a long time. We question whether a routine task in this preparation pipeline, e.g. segmenting larger elements to unit glyphs, can be put as a perceptual task for non-experts.

In this chapter, we first motivate crowdsourcing approaches as alternative solutions to time-intensive tasks done by experts. Then, we briefly discuss the related work on crowdsourcing of large datasets in literature. Later, we describe two crowdsourcing studies on detecting individual Maya glyphs in glyph-blocks. The first study investigates how non-experts perceive the glyph borders in a glyph-block with no or minimal supervision. Based on the conclusions of the first study, in a second crowdsourcing study, we provide more supervision to the non-experts with an improved task design. Furthermore, we analyze different parameter settings leading to satisfactory outcomes before launching a large-scale crowdsourcing job. In this way, a high-quality individual Maya codical glyph dataset (over 9000 glyphs) is produced by the crowdworkers.

The contributions presented in this chapter were originally accepted or published in the following papers:

- Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Is that a jaguar? Segmenting ancient Maya glyphs via crowdsourcing. In *ACM International Workshop on Crowdsourcing for Multimedia*, pages 37–40. ACM New York, November 2014. doi: 10.1145/2660114.2660117

- Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Maya codical glyph segmentation: A crowdsourcing approach. Research Report Idiap-RR-01-2017, Idiap, January 2017c (accepted for IEEE Transactions on Multimedia)

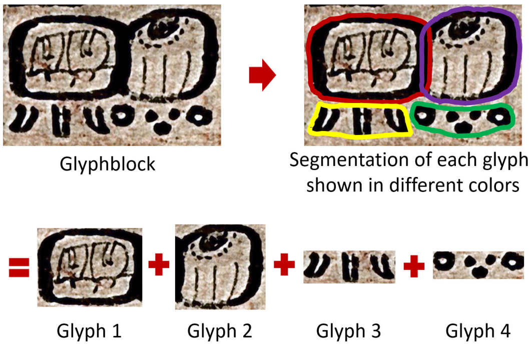


Figure 4.1 – Illustration of the segmentation of individual glyphs out of a glyph-block.

4.1 Introduction

Crowdsourcing is an active area in multimedia to generate labels for images and videos [Biel and Gatica-Perez, 2013; Larson et al., 2012; Nguyen and Gatica-Perez, 2016; Rudinac et al., 2013; Siahaan et al., 2016]. Tagging images, marking object boundaries, and describing scenes or actions are use-cases for image understanding tasks that require large-scale, collaboratively-collected datasets, e.g. ImageNet [Russakovsky et al., 2014] and MS COCO [Lin et al., 2014]. Strategies based both on games [Von Ahn and Dabbish, 2004] and monetary rewards using platforms like Amazon Mechanical Turk (mTurk) have demonstrated their utility in producing labeled image sets of *adequate reliability* for a variety of generic content labels in natural images, including objects, actions, and scenes [Larson et al., 2012].

Similarly, optical character recognition and historical document transcription have advanced thanks to the availability of large-scale datasets like MNIST [LeCun et al., 1998b], IAM [Fischer et al., 2010; Liwicki and Bunke, 2005], and many individual transcription projects [Gatos et al., 2014]. In this perspective, crowdsourcing has been successfully used to produce linguistic resources of historical and cultural heritage materials, e.g. the re-captcha paradigm to transcribe old documents, using a combination of automated document analysis methods and human intelligence [Von Ahn et al., 2008]. Several decades of the New York Times archives have been digitized in this way. Crowdsourcing is also finding other applications in Digital Humanities, in the different phases of dataset generation process, such as scanning documents (digitization phase), locating regions of interest, adding digital entries (transcription phase), and verifying other contributors' responses (correction phase) [Carletti et al., 2013a].

To support the development of multimedia analysis methods for Maya shapes, one basic pressing need is the generation of segmented and labeled glyph data. From a task perspective, glyph segmentation (illustrated in Fig. 4.1) is more challenging than labeling or segmenting natural images due to the following factors:

- **Unfamiliarity.** The participating crowd might have never seen an ancient writing system before, whereas humans interact with and learn about their surroundings from an early age, and have an intuition for object categories (even unseen ones) based on the similarities to already known objects.

- **Visual Complexity.** The Maya language can be visually complex compared to other ancient writings. For instance, Egyptian hieroglyphs are usually in the form of well-separated glyphs. In Maya writings, glyph boundaries are shared between neighbors, the signs can exhibit many deformations, and some inner details are not always visible.
- **Uncertainty.** There are uncertainties about the categories of some signs due to severe damage, incomplete understanding of the changing shape of signs across different eras and places, and unclear semantic relationships of non-frequent signs.

Hence, segmented glyphs are typically produced by experts (i.e. scholars in Maya epigraphy). This is a very time-consuming task, and often tedious for highly-trained scholars. Therefore, we are interested in developing crowdsourcing techniques for ancient Maya hieroglyphics present in digital images from a variety of vestiges (e.g. monuments, ceramics, and codices).

First crowdsourcing task. At their core, glyphs are *visual patterns* that often resemble known objects like animals, human body parts, etc. *One could wonder whether the general human ability to recognize visual patterns could be used for a relatively simple task, namely locate individual glyphs within a glyph-block, with no previous training.* In other words, given a single glyph-block and using only perceptual information, could people guess the number of glyphs and draw bounding boxes around them? If feasible, this could provide a cost-effective alternative for collecting annotation labels for simple tasks. This research question is the basis of our first crowdsourcing task (Section 4.3). We also investigate whether non-experts would generate higher quality outcomes in the case of supervision (i.e. when the number of glyphs in a block is provided).

Second crowdsourcing task. With the same motivation, to study automatic algorithms to analyze Maya glyph shapes, in the second crowdsourcing task, we design a similar glyph segmentation task. In this case, the goal is to build a *Maya individual codical glyph* database from the remaining codex resources (see Section 2.5 for details). According to the observations from the first task, we simplify the task and provide more supervision to the crowdworkers. The task is defined as marking *individual* glyph regions within glyph-blocks (see Fig. 4.1) given the set of variations of each glyph sign contained in these blocks as a means of supervision. These glyph variations are obtained from the existing Maya catalogs created by experts [Macri and Vail, 2008; Thompson and Stuart, 1962] (see Section 2.3 for details). This task design is possible as the textual annotations of the glyphs and the scanned images of the codices have been previously produced by experts.

Finally, let us note that crowd-engagement is a challenge while curating large-scale datasets. Many large-scale digitization/transcription projects are voluntary, due to the lack of resources and vast amount of documents. An alternative approach is to leverage crowdsourcing platforms such as Amazon Mechanical Turk or Crowdflower. These two approaches differ in terms of motivation and engagement of the annotators, the number of annotators available and, in general, the amount of time needed to achieve the annotation task. With paid crowdsourcing platforms, the annotation period is generally shorter, as the crowd is gathered by the platform, and the monetary motivation is the driving force. Hence, careful task design and detailed annotation analysis are required to obtain satisfactory outcomes. Therefore, in our second crowdsourcing task, we employ a gradual experimentation with three stages. In the first stage, we improve the task design, whereas in the second stage we evaluate the performance of the non-experts with respect to several task settings. Based on the annotation analysis on the second stage, we proceed to the last stage to collect a large amount of annotations.

The rest of the chapter is organized in four sections. Section 4.2 describes the related work in literature about crowdsourcing with the goal of large-scale dataset generation. Section 4.3 explains our first crowdsourcing study that assesses the non-expert perception on Maya glyphs. Section 4.4 presents the gradual experimental setup and the detailed annotation analysis in our second crowdsourcing task. Finally, Section 4.5 concludes the chapter by listing the main outcomes from both crowdsourcing studies.

4.2 Related Work

Below, we list several successful crowdsourcing cases, before discussing the main challenges related to the task design, and the resulting annotation reliability.

Large-scale Crowdsourcing Tasks in Multimedia and Computer Vision. Several widely-used benchmarks have been produced via crowdsourcing for recognition, detection, segmentation, and attribute annotation tasks. We can list as example ImageNet [Russakovsky et al., 2014], Microsoft Common Objects in Context (MS COCO) [Lin et al., 2014], SUN scene dataset [Xiao et al., 2010], SUN attribute dataset [Patterson and Hays, 2012], and Caltech-USCD Birds-200 dataset (CUB-200-2011) [Wah et al., 2011]. These large-scale datasets enable to train more capable models in multimedia and vision.

Crowdworkers motivated by monetary rewards (in crowdsourcing platforms) as well as volunteers have been able to generate adequate quality of content for generic object, scene, and action recognition. There has been further crowd content generation studies in sketch recognition [Eitz et al., 2012a] and even in specialized areas such as biomedical imaging [Gurari et al., 2014, 2015; Irshad et al., 2015] and astronomy [Fortson et al., 2012].

Task Design. Gottlieb et al. [2014] discuss the key elements in designing crowd tasks for satisfactory outcomes even for relatively difficult tasks. They emphasize the importance of clear instructions, feedback mechanisms, and verification by qualified annotators. Therefore, in both of our crowdsourcing tasks, we provide clear instructions with positive and negative examples, and give a chance to annotators to provide feedback.

A typical crowdsourcing task follows an "annotation-verification-correction" scheme. However, it may be challenging to apply this scheme to segmentation tasks [Branson et al., 2010]. Especially, in our case, the annotators may not be familiar with the hieroglyphic signs or their perception of the shapes may differ substantially, as the crowd has not been exposed to such visual data as often as everyday life objects in natural images. In order to guarantee satisfactory outcomes, the verification step may require an expert. Therefore, in our second crowdsourcing study, we performed the verification step.

Sorokin and Forsyth [2008] discuss that the commonly-used LabelMe interface [Russell et al., 2008] (and the resulting annotations of everyday objects from natural images) may not generalize to any kind of project specifications. Aligned with this point, we considered to design a more constrained task-specific interface in our first crowdsourcing study. Since our data is not so trivial to mark for non-expert people, first, we have conducted a study about perception of glyphs while giving a range for glyph complexity, rather than the actual number and directly asking people to annotate them. We also want to understand how non-trained eyes would see part-whole relations in archaeological objects.

Su et al. [2012] proposed a system for bounding box annotations around objects in a subset of ImageNet data with 3 steps: drawing, quality verification, and coverage verification. They design these three steps as different tasks on Amazon Mechanical Turk. However, in our first crowdsourcing study, this approach is not applicable to our data where the naive user is not sure even about the glyph complexity and how to divide or merge components to meet the range constraint. In order to guarantee the coverage and quality of the annotations, we have introduced some geometric constraints on the bounding box annotations. For quality, we have made sure that the user cannot select a very small area which would not contain a whole glyph. Second constraint about quality is the overlap ratio. Ideally, bounding boxes of glyphs should not intersect more than a certain amount. By putting this constraint, we aim to eliminate random annotations. We also introduce two different coverage constraints. First one is due to the nature of the experiment, we do not allow the user to submit results where number of total bounding boxes are outside of the given range. Second coverage constraint is a minimum ratio of image coverage by the union of bounding boxes. This ratio is in terms of pixels and not in terms of number of bounding boxes, which is different than Su et al. [2012]'s coverage definition.

Crowdsourcing in Digital Humanities. Digitization and transcription of historical documents with the help of crowdworkers is a widely-studied task in Digital Humanities. A well-known application of this task is the "re-captcha" paradigm that utilizes automated document analysis methods while keeping human intelligence in the loop [Von Ahn et al., 2008]. Several decades of the New York Times' archives have been digitized in this way. In similar transcription tasks [Carletti et al., 2013b; Causer and Terras, 2014], and in archaeological research on a participatory web environment [Bonacchi et al., 2014], crowdsourcing enabled to bring valuable historical sources to the digital era for better preservation of cultural heritage as well as for further analysis.

4.3 First Task: Segmenting Maya Blocks with Minimal Supervision

In this section, we investigate whether reliable annotations of non-experts can be generated as a crowdsourced, glyph-block segmentation task. To our knowledge, this question is novel both in computer science and in digital humanities. For this, we developed an interactive interface and used Mechanical Turk as platform. We use a new data set of glyph-block line drawings for which ground-truth segmentation exists in terms of number of glyphs and their location, which allows to objectively assess the performance of non-experts. We use best practices in Mechanical Turk (regarding requirements for workers and monetary incentives) to recruit workers, controlling for an inherent measure of task complexity (the number of glyphs in the block N_b). Based on the crowdsourced results involving both a pilot study with known workers and a full study with mTurk workers, we show that the task is feasible for glyph-blocks of moderate visual complexity (defined by N_b), and that visual complexity has a clear effect on segmentation performance, measured objectively with respect to the groundtruth. Our framework is overall promising.

4.3.1 Overview of Our Approach

We conduct two studies, a pilot study and a Mechanical Turk (mTurk) study. In both studies, during the annotation task, for a given block, workers have to provide (1) the segmentation of each glyph as a bounding box, (2) a perceived number of glyphs in the block, and (3) the rating of the task difficulty. The annotations are analyzed with respect to: 1) task difficulty, 2) range of perceived number of glyphs and 3) segmentation performance by comparing the number of bounding boxes and their location with the ground truth. Accuracy and purity measures of the crowdsourced segmentations are examined both block-wise and worker-wise.

4.3.2 Data Description

In this section, we used the line drawings generated from the stone monuments in Yaxchilan, an archaeological site located in the state of Chiapas in Mexico. The data consists of drawings of glyph-blocks present in monuments, depicting the visual content with high fidelity, as it is the case with the monumental glyph data described in Section 2.4.

In order to keep the annotation task feasible, we have selected glyph-blocks having 3, 4 or 5 glyphs. Note that this range accounts for the majority of blocks in the datasets we currently work with and constitute a measure of visual complexity.

Segmentation of glyphs in these blocks can be quite challenging for non-experts due to erosion, occlusions, and the inherent visual richness of the glyphs themselves. In this work, we have not used severely eroded blocks. Figure 4.6 illustrates three block examples. The leftmost column corresponds to the ground truth, and from top to bottom, 3-, 4- and 5-glyph examples can be observed. We use a total of 50 glyph-blocks, 31, 12, 7 for 3-, 4-, and 5-glyph cases.

4.3.3 Crowdsourcing Task

We developed a user interface for bounding box annotation, comprising three parts: training, drawing, and evaluation.

Training

To train the workers, we provide clear guidelines, a how-to video (http://youtu.be/WDEmubaF2x0), and examples for each category. The how-to video gives a brief introduction to the Maya writing system, and how to use the interface. To be clear about the task, we also provide a few positive and negative examples. Obviously, bounding boxes covering very small areas are not desired. Negative examples also include cases of too-much-overlap and not-enough-image-coverage. Our goal is that after these guidelines, workers will rely on their perceptual skills.

Drawing

In the main drawing pane, the worker clicks on one edge to start drawing a bounding box and ends by clicking on the diagonal edge. The worker can also remove bounding boxes. The main pane also provides information about the expected block complexity expressed as a range for the number of glyphs in the blocks. This is a key piece of prior knowledge to focus the human task on a narrower set of possible answers. At the same time, it reflects the natural statistics of glyph-blocks.

Evaluation

We also ask the workers to rate the difficulty of segmenting the block (in a scale of 5) and to declare the approximate number of glyphs they would have provided if we had not specified it a priori, namely less than 3, between 3 and 5, and more than 5.

Worker Population

Given the novelty of the segmentation task, we decided to first conduct a pilot study with a smaller set of glyphs and workers we personally knew before launching the mTurk study. The first pilot study has 15 participants and 30 glyph-blocks, whereas in mTurk study there are 10 annotators per block and 50 glyph-blocks. In the pilot task, 3-, 4- and 5-glyph-blocks have the same number of examples (10) each. However, in the mTurk study, as we have selected blocks with catalog annotations, the number of glyphs per block category is 31, 12 and 7 respectively. In the pilot task, participants are not paid, however they are committed and reliable sources. In the mTurk study, we limited our crowd to the ones with *master*'s level expertise and an acceptance rate of at least 95%. In terms of time required to collect the annotations, for the pilot study it took approximately 10 hours to get responses from all participants, whereas for

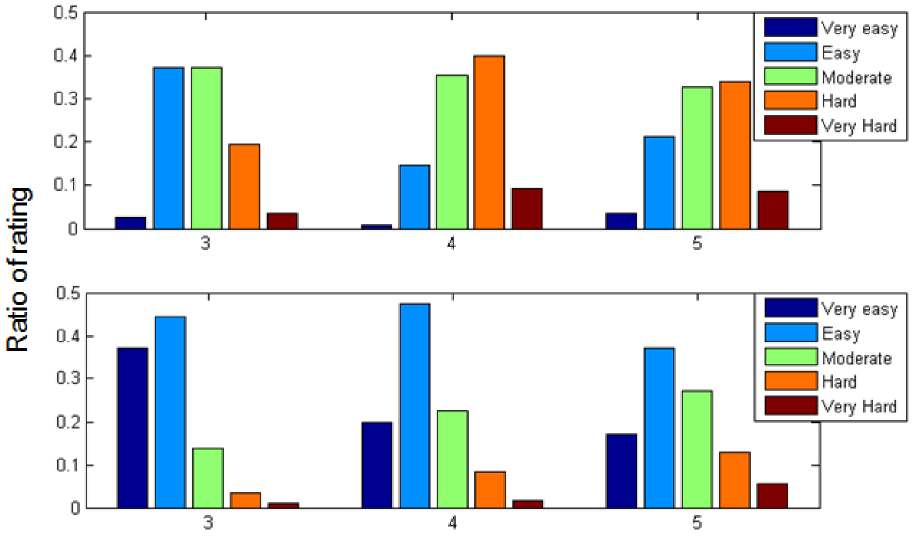


Figure 4.2 – Task difficulty from pilot study (top) and mTurk study (bottom) for 3-glyph, 4-glyph, and 5-glyph-blocks.

mTurk study it took around 2 hours. The estimated task duration is around 1 minute. Each mTurk HIT was paid at 0.15 USD.

4.3.4 Results and Discussion

In this part, we analyze the crowdsourced data from three perspectives: task difficulty, glyph range perception, and segmentation performance. For the pilot study, we have 450 annotations for 30 blocks; for the mTurk study, 500 annotations for 50 blocks from 23 unique workers.

Task Difficulty

For this analysis, the explicit ratings of the workers about the drawing task are evaluated. Figure 4.2 plots the relative proportion for 3-, 4-, and 5-glyph cases. Interestingly, in the mTurk study, workers tend to mark the task easier than in the pilot study. On the other hand, as the number of glyphs increase, the increasing trend of hard and very hard ratings remains similar in both studies. We can conclude that 3-glyph cases are considered easier than 4-glyph and 5-glyph cases.

Range Analysis

For this question (number of glyphs guessed without any constraints), the distributions are plotted in Figure 4.3. Although it can reasonably argued that people were biased towards the suggested range, workers still choose out-of-range options and the plots still indicate a decreasing trend of "in the range of 3-5" as blocks get more complex, more noticeably in the

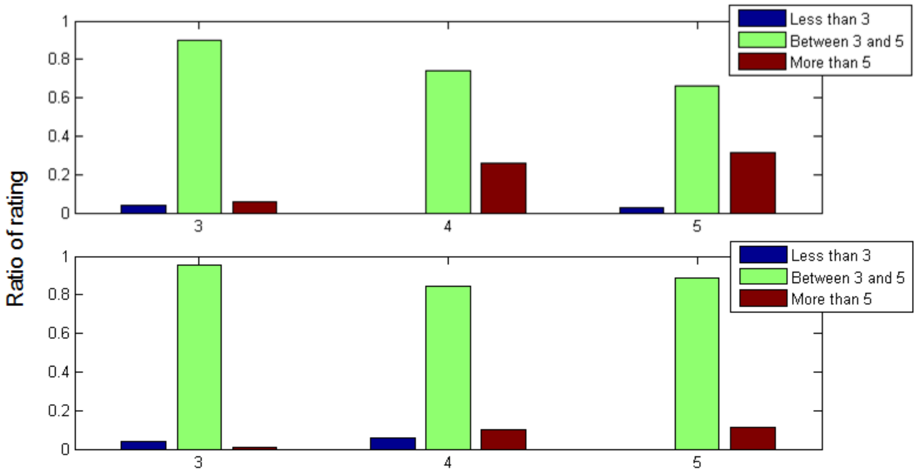


Figure 4.3 – Proportion of perceived number of glyphs from pilot study (top) and mTurk study (bottom) for 3-glyph, 4-glyph, and 5-glyph cases.

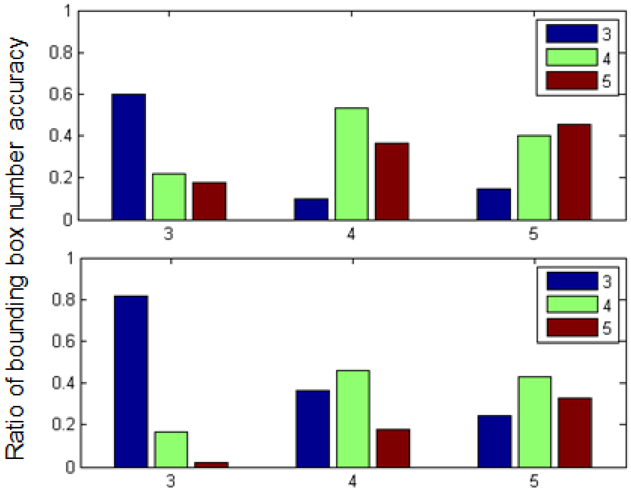


Figure 4.4 – Percentage of bounding boxes from pilot study (top) and mTurk study (bottom).

pilot study.

Segmentation Performance

Segmentation annotations are studied in two aspects: number of bounding boxes and area-wise comparison of segmented vs. ground truth bounding boxes.

Bounding Box Number Analysis. As observed from Figure 4.4, there is a decreasing trend in the correct number of bounding boxes as glyph complexity increases. This is expected, since people get more confused about marking more complex glyphs. Interestingly, the mTurk workers did a better job for the 3-glyph case (0.8 vs 0.6).

Area-Based Performance Analysis. To measure the objective performance of the bounding box annotations, two metrics (accuracy and purity) are used:

$$accuracy(A,G) = \frac{1}{N_o}\sum_k max\left(\frac{|a_k \cap g_{j_k}|}{|a_k \cup g_{j_k}|}\right) \tag{4.1}$$

$$purity(A,G) = \frac{\sum_k max_j\,|a_k \cap g_j|}{\sum_k |a_k|} \tag{4.2}$$

where $A = \{a_1, a_2, ..., a_k, ..., a_n\}$ is the set of the annotation bounding boxes of a worker for a glyph-block, and $G = \{g_1, g_2, ..., g_k, ..., a_n\}$ is the set of ground truth bounding boxes for that glyph-block. Correspondence between an annotated bounding box a_k and a ground truth bounding box g_j is found by $j_k = \underset{j}{argmax}\left(\frac{|a_k \cap g_j|}{|a_k \cup g_j|}\right)$. N_o stands for the number of annotated boxes who suffice an overlapping constraint.

For accuracy, the mean intersection over union ratio of annotation and ground truth bounding boxes is computed. With this measure, we penalize sloppy annotations. Equation 2 is the well-known cluster purity measure [Manning et al., 2008] defined over bounding box regions. These two measures are correlated by a factor of 0.61 in the mTurk data.

Block-based Analysis. In the mTurk study, high performance values are obtained, however the mean values decrease and the standard deviation increases as blocks get more complex (see Table 4.1). Figure 4.5 shows the accuracy and purity of mTurk annotations. As Table 4.1 and Figure 4.5 show, blocks with fewer glyphs are segmented more accurately, with highest values of 0.82 and 0.95 for accuracy and purity for the 3-glyph case.

In Figure 4.6, the first row shows the best case of annotation based on accuracy values (where annotators guessed correctly). The other rows show the worst annotations. The bottom row is a 5-glyph-block where workers get confused about the glyphs on the top as well as whether to merge the small elongated glyphs on the lower part with the head-like shapes on top of them. We can also see this merging tendency issue of elongated glyphs with the head-like glyphs in the second example. We can also observe that worker 2 marked three circles on the left separately, probably because they are well separated, and mark the rest complex part as one.

About the annotations, we observe that they are in good quality in general. For instance in the bottom row, only worker 4 has not left an unmarked closed contour on the upper right part. We encountered very few sloppy bounding boxes and no random marking at all. We hypothesize that the coverage and overlap constraints on the user interface helps increase the

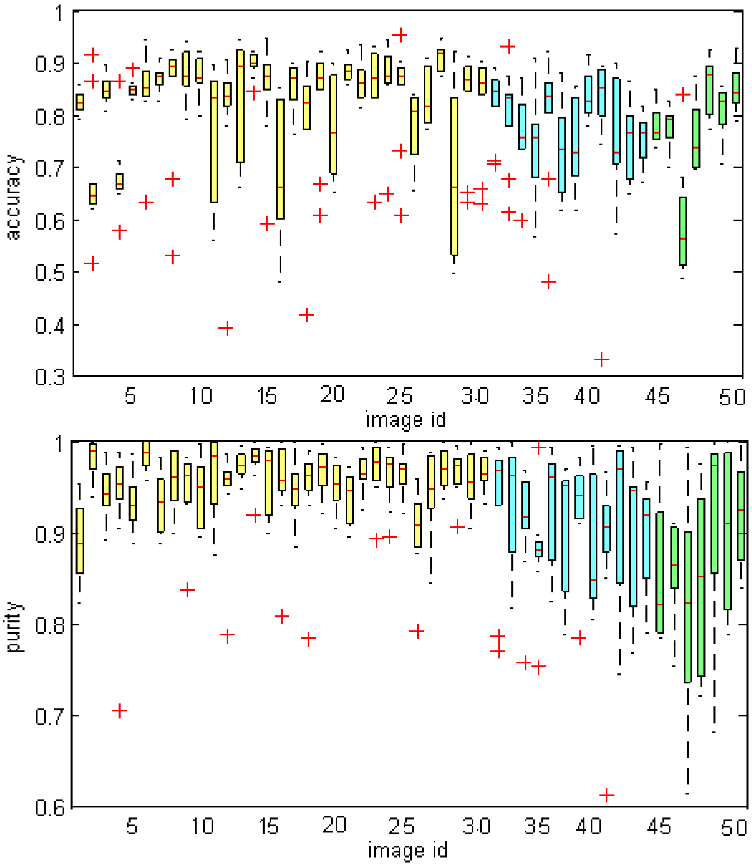


Figure 4.5 – Block-based annotation accuracy (top) and purity (bottom) from mTurk study. Yellow: 3-, blue: 4-, green: 5-glyph-blocks.

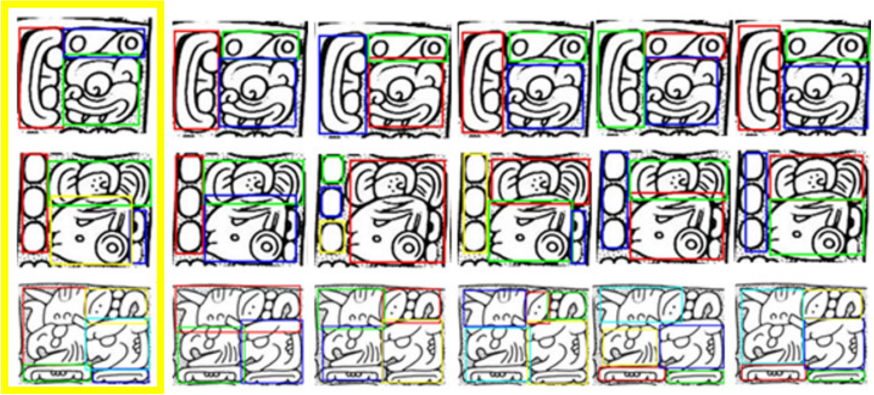


Figure 4.6 – Top two and bottom scored image annotations from mTurk. The first column is the ground truth, and other columns are the annotations of 5 workers. Drawings produced by Graham and Von Euw ©[Graham, 1979; Graham and Von Euw, 1977], block segmentation and glyph annotations provided by Carlos Pallán Gayol. Visualize in pdf for details.

Table 4.1 – Block-based annotation performance for mTurk study for 3-glyph, 4-glyph, and 5-glyph cases.

	Mean Acc.	Std. Acc.	Mean Pur.	Std. Pur.
3	0.820	0.063	0.951	0.020
4	0.725	0.047	0.907	0.022
5	0.692	0.103	0.871	0.042

high performance values.

Worker-based Analysis. Performance for each worker is shown in Figure 4.7, where workers are ordered based on the number of blocks they annotated (shown as percentages on top of the bars). We observed that some workers marked only a few blocks, which is typical in crowdsourcing. Their performance is sometimes better than the few workers who worked almost all of the blocks as the latter must have encountered hard cases in the dataset as well (and possibly experienced fatigue). Average accuracy per worker ranges between 0.64 and 0.92 as purity is between 0.88 and 0.98.

4.3.5 Conclusions of First Task

We presented a new use of crowdsourcing for generating segmentations of ancient Maya glyphs by non-experts. The task was designed as a constrained segmentation problem with little prior training, and that largely relied on perceptual organization skills of workers. Using a variety of segmentation quality measures, we conclude that the task is feasible for moderate visual complexity (measured by the number of glyphs in a block), and that less complex blocks (containing 3 glyphs) were indeed easier than other cases.

Given the formidable challenges of the Maya script, by no means we claim that the crowd can substitute expert knowledge in epigraphy. Rather, the results suggest that non-expert work could be useful for simple, well-designed segmentation tasks, which could later be verified by experts.

In the next section (Section 4.4), in addition to using a significantly larger data set, we investigated whether more accurate segmentations can be obtained with variations of the task presented here, e.g. by modifying the interaction paradigm or adding information coming from extra sources like glyph catalogs.

4.4 Second Task: Generating Individual Glyphs For the Three Ancient Codices

The focus of this section is on producing individual glyph shape data from the three original Maya Codices (Dresden, Madrid, Paris) via online crowdsourcing. We present our design of

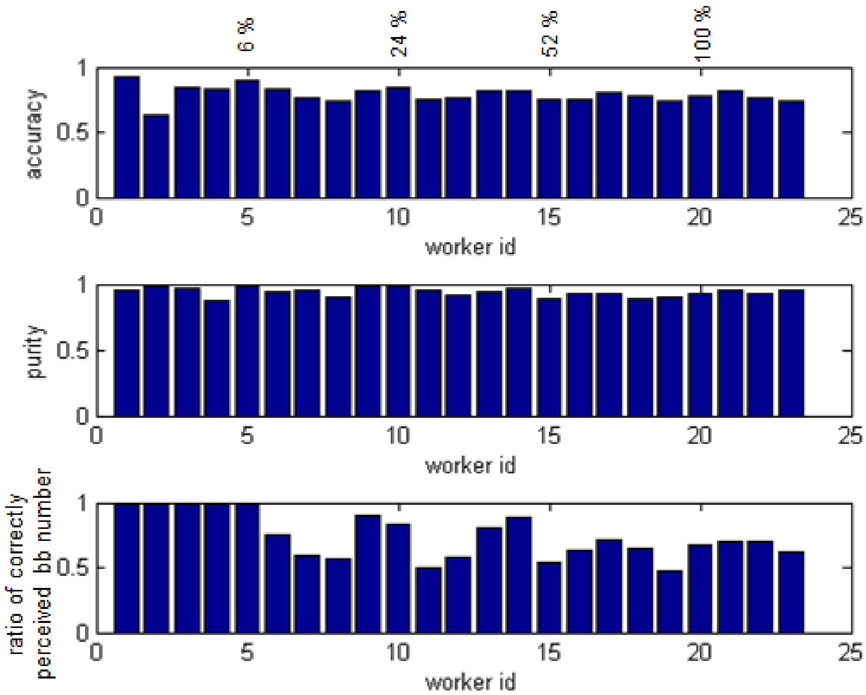


Figure 4.7 – Worker-based accuracy (top), purity (middle), and ratio of correctly perceived glyph number (bottom) from mTurk study.

the crowdsourcing task, investigating the effects of several features like the task definition, the use of different classic catalogs (Thompson and Macri-Vail) as glyph pattern models, and the relationship between the number of annotators, the sample complexity, and the reliability of the generated ground truth.

In summary, the contributions of this section are two-fold:

1. *Glyph segmentation crowdsourcing:* Novel task accounting for fine-grain mapping of catalog variants to codex samples, and multi-way assessment of outcomes.
2. *Dataset curation and creation:* Construction of a new, segmented 9000 glyph dataset that has been made publicly available. To our knowledge, this will be the largest public database of individual Maya glyphs.

From our experiments, we observed that in spite of the glyph complexity, two non-expert annotations are enough in the majority of the cases to produce a consensual segmentation: For around 85% of the glyph cases, two contributors agree on the marked glyph area (overlapping more than 80%). We show that the contributors were confused and failed to reach a consensus in few cases that are challenging due to damaged target glyphs or other similar regions in the glyph-blocks.

The rest of the section is organized in four parts. Subsection 4.4.1 explains the design and evolution of our crowdsourcing task. In Subsection 4.4.2, the details of the experimental procedure are provided. In Subsection 4.4.3, the annotations are analyzed with respect to key aspects in the pipeline. Finally, Subsection 4.5 concludes the section.

4.4.1 Crowdsourcing Task

Automatic glyph recognition starts with obtaining segmented, cleaned, and binarized glyph data. We investigated whether the first part of this preprocessing task (glyph segmentation) can be crowdsourced. In our work, non-experts were asked to segment individual glyphs from the original glyph-block sources. Our experimental design evolved over three stages (**preliminary, small, large**). In the preliminary stage, we segmented few glyphs (27 from randomly-chosen 10 blocks) with two different task designs. This stage helped to define a final task design. The small stage consists of segmenting glyphs that have ground truth (a subset of glyphs from [Hu et al., 2015]). This stage helped to judge which catalog was more helpful to non-experts in our task. At the large stage, we conducted the segmentation task for over 10K glyphs.

In this subsection, we explain the process that led to the design of the final task. First, we describe the requirements and present the platform used for experiments. We then discuss the early experience on the task design. We finally describe the final version of the task.

Requirements

Given the annotations in the glyph-blocks (provided by epigraphy experts), and the example sign variants (taken from the catalogs), we expect crowdworkers to segment each individual sign in a block. As Maya glyphs can be found in articulated forms, i.e. hand signs, cropping glyph regions via bounding boxes may end up with inclusion of some parts of the neighbor glyphs. Therefore, for better localization, we designed the segmentation process to be done as free-polygons rather than bounding boxes.

To guide the process, we show workers the different variants of the sign to be segmented. As validation information, we would like to know what sign variant the annotator chose as template to segment each glyph, and how similar the chosen variant and the marked region. This can be used to verify the expert annotations and detect outliers, in case when none of the provided sign variants match the block content. To account for this, we propose a "None" option along with the existing sign variants.

Another point to analyze is the perception of damage by non-experts. Even though experts have provided a damage score for each glyph, this score shows how decipherable the glyph is, and so it is affected by the glyph co-occurrence and semantics. Non-expert perception of damage depends solely on visual appearance. This helps to obtain a damage score that is not affected by prior expert knowledge. The score can also be used as a hint to assess the task difficulty.

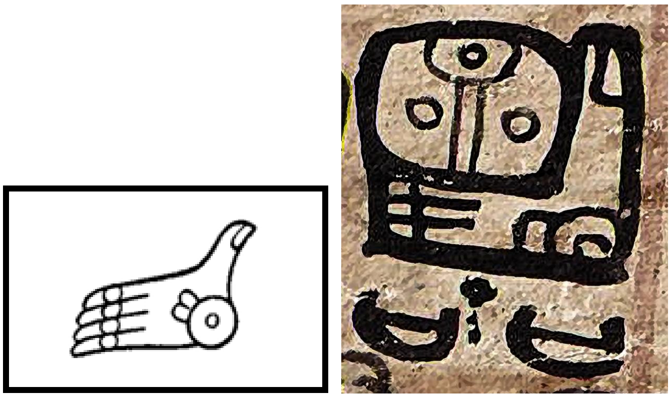


Figure 4.8 – An articulated hand sign (T670) from Thompson catalog and an instance of it from Paris codex. This example shows the requirement of polygon segmentation in the task design.

The difficulty of our task is not uniform across categories. According to the visual similarity to the variants and the damage of the glyph, the task can be ambiguous. To assess this, we ask workers to provide a score for the task difficulty.

Platform

Terminology. We utilized the Crowdflower (CF) platform for our experiments. In CF terminology, a *job* refers to the whole annotation process. An annotation unit is called *task*. A *page* is a set of unit tasks that a contributor needs to complete to get paid. N_t denotes the number of tasks in a page. The number of judgments per task N_j corresponds to the number of workers that should annotate a single task. Workers in CF are called *contributors*. There are three levels of contributors. The level of a contributor is based on the expertise and performance in previous tasks.

To set up a job, a job owner must first define the dataset to be annotated. The job owner designs the task by specifying the queries that the contributors are asked to complete. The queries in the task can vary from simple text input to performing image annotations. After the task design is finalized, the job owner can curate *test questions* (TQ) to enable the *quiz mode* in the job to ensure the quality of the results. Test questions are prepared by the job owner by listing acceptable answers for each query in the task. If the contributor gives an answer out of the acceptable answers, the contributor fails the test question. For the image annotation query, the job owner provides a ground truth polygon over the image and sets a minimum acceptable intersection-over-union (IU) threshold. The IU measure between segment S and ground truth G is defined as follows:

$$IU = \frac{|S \cap G|}{|S \cup G|}. \tag{4.3}$$

If a contributor marks a region whose overlap with the ground truth region is below the IU

Table 4.2 – Preliminary stage segmentation results using variants of Thompson catalog (T).

Exp.	**Block-based or glyph-based**	**# Judgments per task (N_j)**	**# Tasks in a page (N_t)**	**Payment per page ($)**	**Min level of contributors**	**Allowed Channels**	**Average f-measure (%)**
1	Block-based	10	10	0.15	Medium	All	75.2
2	Block-based	5	2	0.30	High	All	79.5
3	Glyph-based	5	2	0.10	High	All except CF-elite	89.7
4	Glyph-based	5	2	0.10	High	CF-elite	92.0

threshold, the contributor fails the test question and cannot take on more tasks in the job. Contributors have to pass one page of the task in quiz mode before being admitted to the *work mode*, in which they work on the actual set of questions (AQ) and get paid. There is also a test question on each page in work mode. This check is effective to eliminate random answers.

The platform provides other quality control checks. Job owners can set the minimum time to be spent on the task, the minimum accuracy that a contributor needs to achieve, and the maximum number of tasks that can be annotated by a contributor. After creating the answers for the test questions and fixing the job settings, the job owner launches the job, and can monitor the progress of the crowd workers.

Channels. CF has its own subscribers, referred to as the Crowdflower-elite (CF-elite) channel. Apart from that, workers from other crowdsourcing platforms (also called channels) can also link their accounts and work on available CF jobs. This allows crowd diversity in the platform. These external platforms can be large-scale, with global subscribers such as ClixSense, or can be medium- or small-scale with a focused crowd in particular countries. The choice of platforms is given to the job owner.

Preliminary Stage: Design Experiences

In the preliminary stage, we conducted four experiments before deciding the final task design and settings. The different settings are given in Table 4.2, and discussed below.

Block-based design vs. glyph-based design. In the first two experiments, the initial design (shown in Fig. 4.9) aimed to collect *all* glyph segmentations of a glyph-block in the same task (one glyph after another in separate drawing panels). This initial design proved to be confusing. Some workers marked all the glyph regions in the first drawing pane, instead of drawing them separately. Another source of confusion was the order of the glyphs. Learning from this, we simplified the task as *individual* glyph drawing. As a result, the average f-measure between the convex hull of a crowd-generated segmentation and the ground truth improved by more than 10% (see Table 4.2), when moving from multi glyph annotations to the single glyph case. More specifically, the f-measure of segment S and ground truth G is defined based on precision p

and recall r as follows:

$$f = 2 * \frac{p * r}{p + r} \quad , \quad p = \frac{|S \cap G|}{|S|} \quad , \quad r = \frac{|S \cap G|}{|G|}. \tag{4.4}$$

Number of glyph variants. We limited the number of glyph variants shown to the contributors to keep them focused on the segmentation task. At first, we experimented with a maximum of three variants chosen a priori by visual clustering (12% of the signs in the Thompson catalog had more than 3 variants). After empirically verifying that increasing the number of provided variants did not hinder worker performance overall, and gave more visual cues about the possible variations, we decided to provide a maximum of six variants (if available).

Design of feedback mechanisms. In the initial design, we asked contributors about glyph damage level as well as wrong or missing annotations. This part was often omitted by the workers. From this experience, we decided to keep only the most direct rating factors (damage and task difficulty). We also included a text box for optional comments. Received comments included remarks about rotations of the glyph variants, uncertainty about the damage rating, and choice of the variants. Based on these comments, we improved the instructions.

Crowd expertise, number of tasks per page, and payment. In the first experiment, we allowed contributors with medium- and high-level of expertise and set the payment per page as $0.15. We hypothesized that 10 tasks per page were too many considering the payment. We observed that only medium-level contributors took the job, and only 60.9% of the glyph segmentations were saved, with an average f-measure of 75.2%. In the second experiment, we decreased the number of tasks per page to 2, set the payment per page to $0.30, and only allowed expert contributors (level-3). This resulted in 79.9% saved segmentations with average f-measure of 79.5%. Considering that there are three glyphs in glyph-blocks in average, we set the payment to $0.10 for the last two single glyph-based experiments to maintain payment/time ratio. Together with the simplified design and the introduction of test questions, this payment and level of expertise brought the saved segmentation ratio very close to 100% (97.3% for the third experiment and 100% for the fourth one) with an average f-measure of around 90%.

Number of judgments. In the first experiment, we started with 10 judgment per task ($N_j = 10$). Based on it, we decided to collect fewer judgments of higher quality. Therefore, we decreased N_j to 5 in the next experiments, and improved the level of expertise and payment settings as explained above.

Crowdflower-elite channel vs. other channels. We experimented with workers from different channels (CF-elite channel compared to other channels) in the last two experiments. With the simplified individual glyph-based design, and with level-3 contributors, we did not experience a significant difference in the segmentation scores from these separate channels (89.7% vs.

Table 4.3 – Experimental settings for the small-scale stage (S-1 and S-2) and the large-scale stage (L-1 and L-2).

Exp.	**Cat. Var.**	**# Judg. per task** (N_j)	**# Tasks per page** (N_t)	**Pay. per page ($)**	**# pages**	**IU th.**
S-1	T	5	2	0.10	338	0.7
S-2	MV	5	2	0.10	344	0.7
L-1	MV	2	4	0.16	1670	0.7
L-2	MV	2	4	0.16	1732	0.8

92%, see Table 4.2). As a consequence, we decided to use all the channels in the following stages.

Final Task

Overview. Based on the outcome of the preliminary stage, we designed the final task comprising two parts (Fig. 4.10). In the first one, based on the shown variants, contributors were asked to segment (draw a tight free-hand polygon) a similar region in the glyph-block. In the second part, contributors were asked to indicate which variant they used as template to do the segmentation, and to rate how similar the variant was to the segmented region, how damaged the glyph region was, and how easy it was to complete the task. These ratings are designed on a scale between 1 and 5.

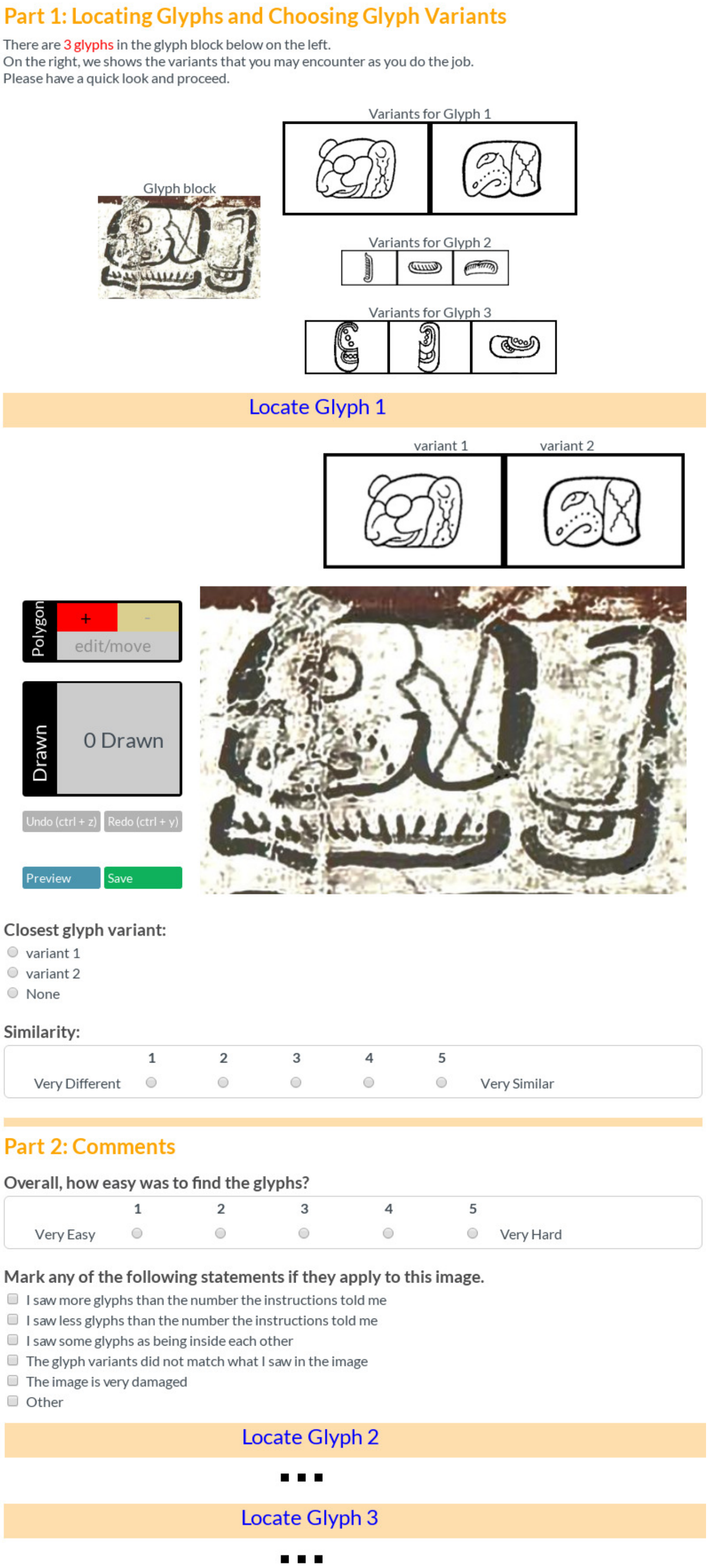
Training. We provided a detailed description of the tasks, a how-to Youtube video, and positive/negative examples of segmentation, example of damage levels, and explained that segmentation quality would be checked.

Drawing. We used the image annotation instance tool in Crowdflower for free polygon drawing over the glyph-block images. This tool allows correction and multiple polygons, which is useful for glyph repetition cases.

Evaluation. We selected the quiz mode for the jobs: we provided tasks with known answers (ground truth polygons) and a quality threshold on intersection-over-union (IU) measure (see Section 4.4.1) to filter out spammers and increase quality.

4.4.2 Experimental Protocol

Given the decisions made during the preliminary stage, we first conducted the small-scale stage over the glyphs which have ground truth, and then we run the large-scale stage. This section explains the settings of these two stages.



Part 1: Locating Glyphs and Choosing Glyph Variants

There are 3 glyphs in the glyph block below on the left.
On the right, we shows the variants that you may encounter as you do the job.
Please have a quick look and proceed.

Locate Glyph 1

Closest glyph variant:
○ variant 1
○ variant 2
○ None

Similarity:

	1	2	3	4	5	
Very Different	○	○	○	○	○	Very Similar

Part 2: Comments

Overall, how easy was to find the glyphs?

	1	2	3	4	5	
Very Easy	○	○	○	○	○	Very Hard

Mark any of the following statements if they apply to this image.
□ I saw more glyphs than the number the instructions told me
□ I saw less glyphs than the number the instructions told me
□ I saw some glyphs as being inside each other
□ The glyph variants did not match what I saw in the image
□ The image is very damaged
□ Other

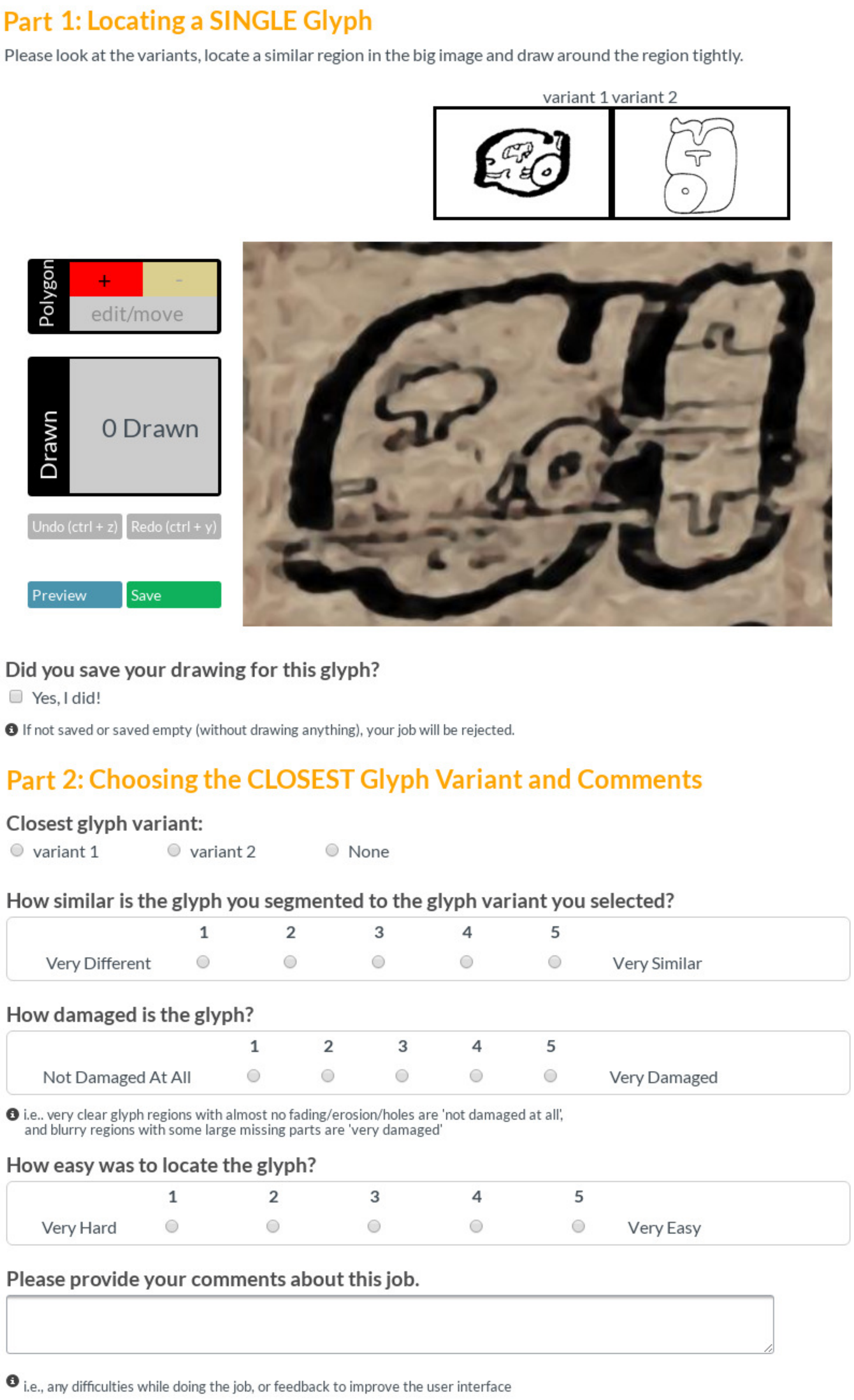
Locate Glyph 2

■ ■ ■

Locate Glyph 3

■ ■ ■

Figure 4.9 – Initial block-based task design, illustrating only the first glyph in the block for brevity.



Part 1: Locating a SINGLE Glyph

Please look at the variants, locate a similar region in the big image and draw around the region tightly.

Did you save your drawing for this glyph?

□ Yes, I did!

ℹ If not saved or saved empty (without drawing anything), your job will be rejected.

Part 2: Choosing the CLOSEST Glyph Variant and Comments

Closest glyph variant:

○ variant 1 ○ variant 2 ○ None

How similar is the glyph you segmented to the glyph variant you selected?

	1	2	3	4	5	
Very Different	○	○	○	○	○	Very Similar

How damaged is the glyph?

	1	2	3	4	5	
Not Damaged At All	○	○	○	○	○	Very Damaged

ℹ i.e.. very clear glyph regions with almost no fading/erosion/holes are 'not damaged at all',
and blurry regions with some large missing parts are 'very damaged'

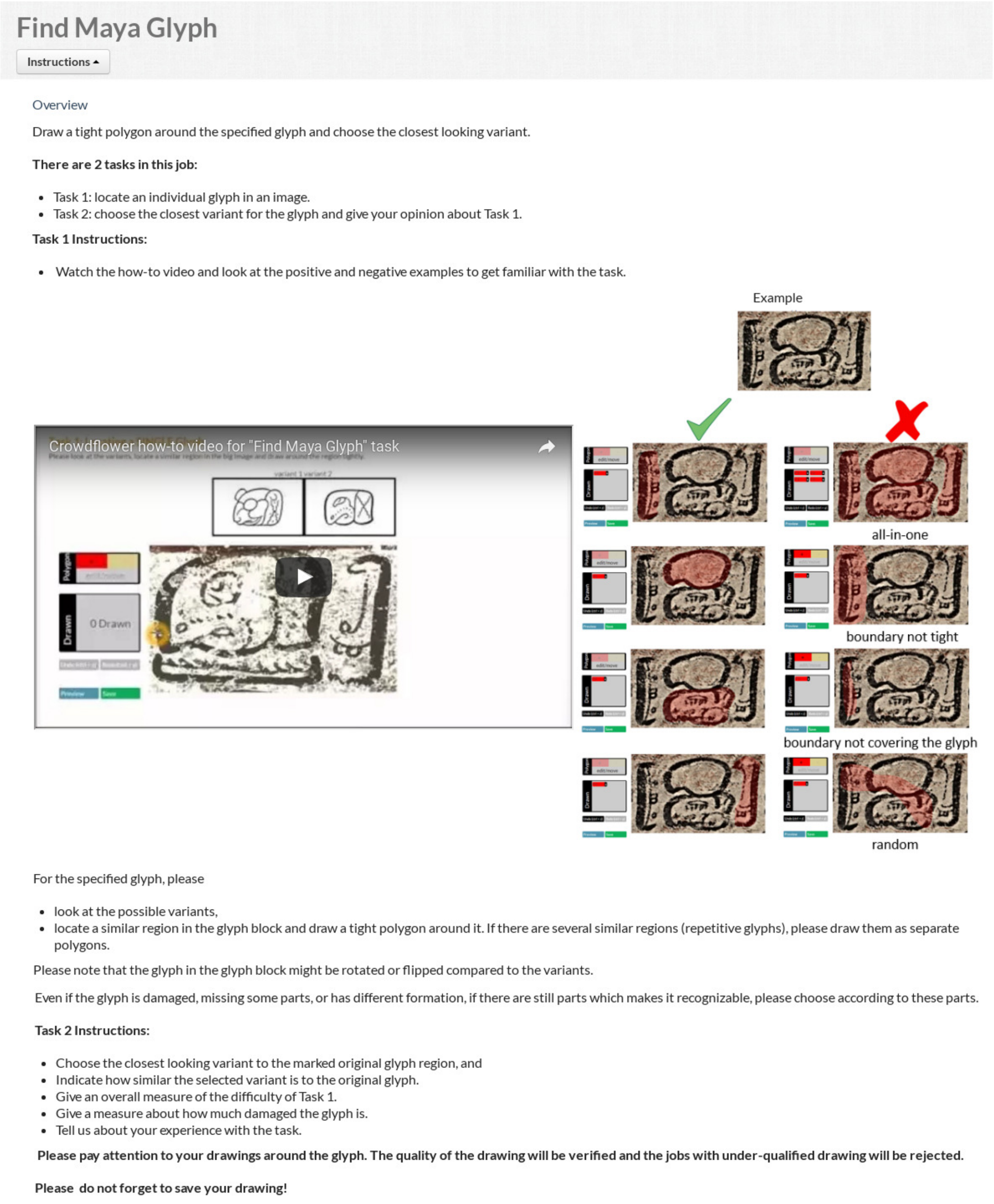
How easy was to locate the glyph?

	1	2	3	4	5	
Very Hard	○	○	○	○	○	Very Easy

Please provide your comments about this job.

ℹ i.e., any difficulties while doing the job, or feedback to improve the user interface

Figure 4.10 – Final task design.



Find Maya Glyph

Instructions ▴

Overview

Draw a tight polygon around the specified glyph and choose the closest looking variant.

There are 2 tasks in this job:

- Task 1: locate an individual glyph in an image.
- Task 2: choose the closest variant for the glyph and give your opinion about Task 1.

Task 1 Instructions:

- Watch the how-to video and look at the positive and negative examples to get familiar with the task.

For the specified glyph, please

- look at the possible variants,
- locate a similar region in the glyph block and draw a tight polygon around it. If there are several similar regions (repetitive glyphs), please draw them as separate polygons.

Please note that the glyph in the glyph block might be rotated or flipped compared to the variants.

Even if the glyph is damaged, missing some parts, or has different formation, if there are still parts which makes it recognizable, please choose according to these parts.

Task 2 Instructions:

- Choose the closest looking variant to the marked original glyph region, and
- Indicate how similar the selected variant is to the original glyph.
- Give an overall measure of the difficulty of Task 1.
- Give a measure about how much damaged the glyph is.
- Tell us about your experience with the task.

Please pay attention to your drawings around the glyph. The quality of the drawing will be verified and the jobs with under-qualified drawing will be rejected.

Please do not forget to save your drawing!

Figure 4.11 – Final instructions of the task.

Small-scale stage

In this stage, we run two experiments whose parameters are summarized in Table 4.3. For the 823 individual glyphs (322 blocks) that have expert ground truth masks, we set up the task with Thompson (T) and Macri-Vail (MV) references of the glyphs. In other words, we display the glyph variants from either the Thompson or the Macri-Vail catalogs.

In both cases, the number of judgments N_j was set to 5. The minimum acceptable IU score was set to 0.7. The minimum time to be spent on a page was set to 30 seconds. The maximum number of judgments by a single contributor was set to 12. As a result, a single contributor annotated 5 glyphs from the actual target set and also answered 7 test questions.

Large-scale stage

In this stage, we define the job for all annotated glyphs for which no expert segmentation is available. To reduce the annotation cost and having confirmed that in general most of the glyphs had a high segmentation consensus (see small-scale stage analysis in Section 4.4.3), we decided to collect only two judgments per glyph, and collect more only if a disagreement was detected. We decided to exclude the following glyphs from the annotation:

- Too damaged glyphs according to the damage scores by the expert and visual post-inspection of a team member,
- Repetition cases (multiple instances of the same glyph in the block),
- Infix cases (two separate glyphs merged by modern decipherment for semantic reasons).

As a result, we obtained 10126 glyphs to be annotated (out of 14722 glyphs from the available segmented glyph-block images). For this stage, we only relied on the Macri-Vail catalog which is a more modern resource in epigraphy.

We set the minimum IU threshold to 0.7 for the first half of the glyphs (5000 glyphs) and 0.8 for the rest. This threshold ensured that the contributors did a good job on the test questions, and presumably on the actual questions, so that high consensus on the collected segmentations for each glyph can be obtained. We observed that we need contributors with higher performance, as we depend on the segmentations coming from only two contributors per glyph in this setting. That is why we increased the minimum IU threshold for the second half of the glyphs. The minimum time spent on the task was set to 30 seconds. The maximum number of judgments by a single contributor was set to 48.

Segmentation Evaluation Procedure

Evaluation was performed by comparing the ground truth of the glyphs with the crowd segmentations for the small-scale stage. This is detailed in Section 4.4.3. For the large-scale stage, we compare the segmentations of the contributors against each other. We also checked problematic cases in which the f-measure agreement was less than 0.8 among contributors as an internal task in Crowdflower platform.

4.4.3 Crowdsourced Annotation Analysis

In this section, the crowd annotations for the small-scale and large-scale stages are presented in terms of the analysis of ratings and segmentations.

Small-Scale Stage

As described in Section 4.4.2, we conducted two experiments in small-scale stage, with Thompson (T), and with Macri-Vail (MV) references of the glyphs. We analyze the annotations from these experiments w.r.t. four aspects: variant selection, damage rating, segmentation analysis, and sensitivity to the number of annotators.

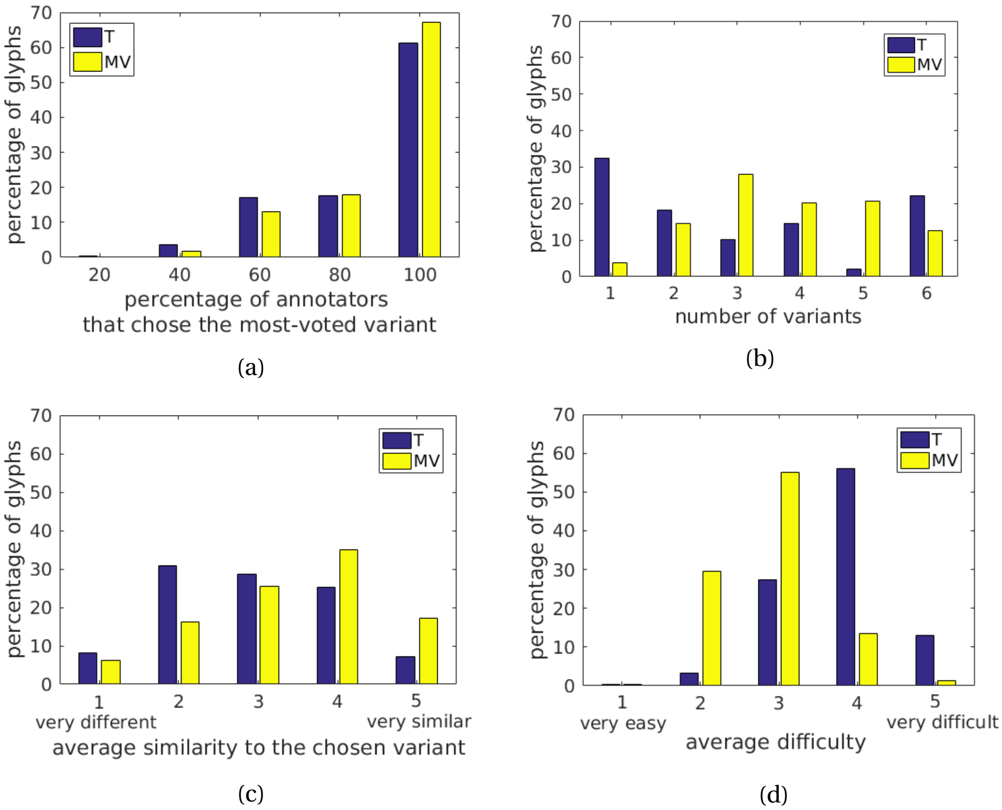
Variant Selection. We compare the agreement for the variant selection in the two experiments. First, note that the MV catalog contains the glyph variants from both codices and monuments, whereas the variants in the Thompson catalog come only from monuments. Typically, monumental glyphs have more details and are visually more complex than codical glyphs. In this sense, the variants from the Thompson catalog are in general more different from the codices glyphs than the MV variants.

The final variant for each glyph was selected by majority voting among the contributors' responses. Fig. 4.12a shows the percentage of contributors that selected the most-voted variant for the experiments with the Thompson (blue) and Macri-Vail (yellow) variants. We observe that all of the contributors agreed on a variant for 67.2% of the glyphs when the MV variants (yellow) were shown (61.2% for the T case).

Fig. 4.12b shows the histogram of the number of variants for the annotated glyph categories. The median values are 2 and 4 for T (blue) and MV (yellow) variants, respectively. Thus, even though there were in general more variants available, for the MV cases full agreement was higher (Fig. 4.12a).

A related result is illustrated in Fig. 4.12c. Contributors gave higher ratings of visual similarity to the MV variants rather than T variants (2.98 vs. 2.46 mean similarity). Moreover, the contributors found the task harder in the case of T variants (Fig. 4.12d). These differences in similarity and difficulty ratings were significant as measured with Kolmogorov-Smirnov non-parametric hypothesis testing [Massey Jr, 1951].

In summary, we observed that MV-variant tasks are rated easier, and reach higher consensus



(a)

(b)

(c)

(d)

Figure 4.12 – Distributions of average ratings in the small-scale stage with Thompson (blue) and Macri-Vail variants (yellow).

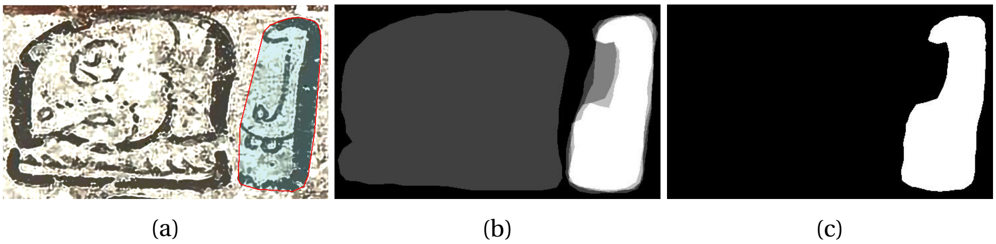


Figure 4.13 – (a) Convex hull of the ground truth for the glyph on the right (red line, blue filling), (b) gray-scale image of the aggregated segmentations, and (c) final aggregated segmentation.

rates than the T-variant cases.

Damage Rating. The average damage ratings (scale 1 to 5) by the crowd and the damage rating assigned by the experts are considerably different. For the experts, more than 90% of the glyphs in this set were easily recognizable (5 in the range 1 to 5). However, the damage perception of the non-experts was focused around the middle of the scale. For 64% of the glyphs, the contributors selected "moderate-damage" (3 in the range 1 to 5) for both T and MV cases. This can be interpreted as the raw block crops being visually noisy in most of the cases, even though for the experts the glyphs are in good conditions to be identified.

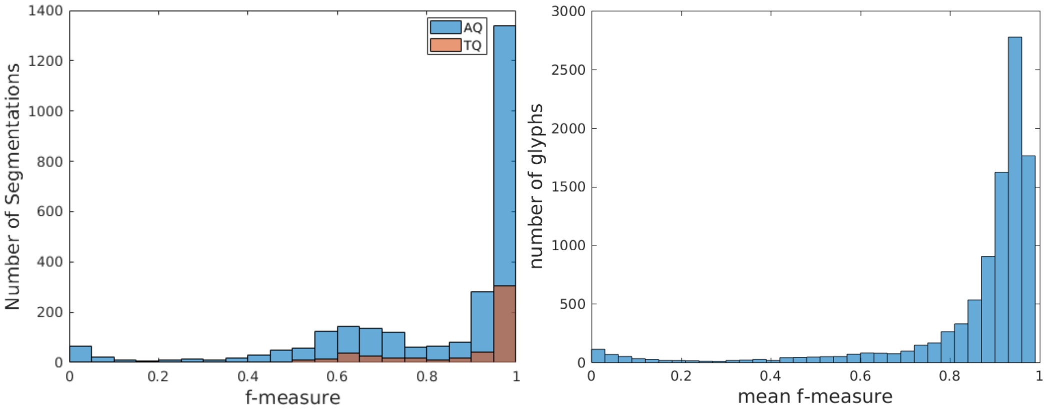
Segmentation Analysis. For each glyph, an aggregated mask is generated from the crowd segmentation masks, such that at least half of the contributors (i.e, at least 3) marked an image point as belonging to the glyph region as illustrated in Fig. 4.13.

The evaluation is performed by comparing

1. the aggregated segment against the binary ground truth (S vs. GT); and
2. the convex hull of the aggregated segment against the convex hull of the ground truth (S-CH vs. GT-CH).

Results are shown in Table 4.4. We observed that most of the contributors mark the glyph regions without going into fine contour details, as it can be quite time-consuming. This is acceptable, as the main interest is in the regions with the target glyph rather than with very detailed contours. Therefore, we decided to use convex hulls for further evaluation in Figs 4.14-4.15.

Table 4.4 summarizes the comparative segmentation performance with the help of the two catalogs. It is observed that the MV variants helped to bring out marginally better aggregate segmentations. The table also reports the mean scores when we consider the glyphs used as test questions (TQ) and actual questions (AQ) as separate sets. The f-measure distributions of TQ and AQ sets in the MV variants cases are plotted in Fig. 4.14 (the T variants case is similar and thus not shown). We observe that the majority of the glyphs are well segmented. As we



(a)

(b)

Figure 4.14 – (a) The f-measure distributions of overlap between crowd segmentations and ground truth in actual question set (AQ, blue) and test question set (TQ, orange) with the MV variants in the small-scale stage. (b) The mean f-measure agreements for the glyphs in large-scale stage.

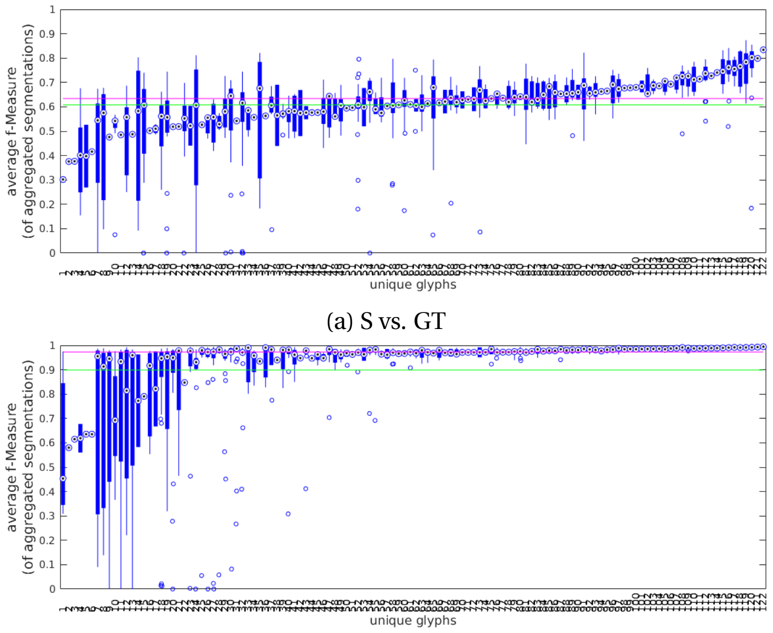
Table 4.4 – Average f-measure values of aggregated segmentations obtained with Thompson (T) and Macri-Vail (MV) variants in small-scale stage for test questions (TQ) and actual questions (AQ).

Catalog Variants	Set	S vs. GT (%)	S-CH vs. GT-CH (%)
T	TQ	65.7	96.6
MV	TQ	65.5	97.3
T	AQ	59.1	87.5
MV	AQ	59.9	88.6
T	All	60.2	89.0
MV	All	60.8	89.9

manually chose the test questions to be relatively easy to annotate, we observe a higher mean f-measure for TQ compared to AQ.

Fig. 4.15 illustrates the boxplots of the sorted average f-score values of 122 non-numerical MV classes (left for S vs. GT, and right for S-CH vs. GT-CH). While most of the classes are well segmented, few of them have low average f-measure (5 classes have an average f-measure less than 40%). We observe that these classes are visually more complex and composed of several parts. When using the convex hull comparison, only ten classes have an average f-score less than 70%.

Sensitivity to The Number of Annotators. We simulated the performance for the case of fewer annotators. Fig. 4.16 shows the average f-measure values for the aggregated masks with different number of segmentations (2-5). We aggregated a maximum of 10 combinations of randomly selected segmentations, and took the mean f-score of these aggregated masks for



(a) S vs. GT

(b) S-CH vs. GT-CH

Figure 4.15 – Sorted average f-measure of aggregated segmentations for the unique glyph categories in the small-scale stage. Green and red lines indicate overall mean and median values, respectively.

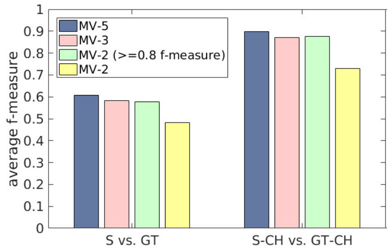


Figure 4.16 – Mean f-measure values of the aggregated masks obtained using 5 (blue), 3 (pink), 2 (yellow) segmentations, and 2 segmentations that have at least 0.8 f-measure agreement (green) per glyph with MV variants.

each glyph. Obtaining aggregated masks with 3 segmentations (MV-3) rather than 5 (MV-5) resulted in a marginal decrease in the average f-score (blue to pink bars).

Furthermore, we analyzed the intersection of two segmentations either for the randomly selected ones (MV-2 yellow bars) or in the case of above 0.8 f-measure agreement (MV-2 green bars). In the latter case, we obtained very similar average f-score results to the ones with 3-segmentations. The standard deviation of the f-measures obtained with randomly sampled 2-annotations are below 0.1 and are usually acceptable. **These observations motivated us to perform the large-scale stage with two annotations per glyph and validate the segmentation when the agreement was higher than** 0.8**.**

Outcome. 368 and 397 unique contributors participated to the small-scale stage for the T-variant and MV-variant cases, respectively. The corresponding average number of glyph annotations per contributor were 7.3 and 8.9 (median 5 and 6, respectively). The evaluation in this subsection shows that the defined task is simple enough for a non-expert to produce satisfactory results. Even though the contributors may get confused, overall the performance was high enough to proceed with the large-scale stage.

Large-Scale Stage

Here, we analyze the results obtained for the large-scale stage. We obtained 21907 annotations containing 20982 saved segmentations.

Glyph Variant Selection. Fig. 4.17a shows that the first variant was chosen in 73.2% of the annotations. This is not surprising as usually the two first variants in the Macri-Vail catalog are instances directly taken from the codices, and the others are drawings of more complex monumental glyphs taken from the Macri and Looper [2003] catalog. In 7.7% of the annotations, the "none of the variants" option was chosen.

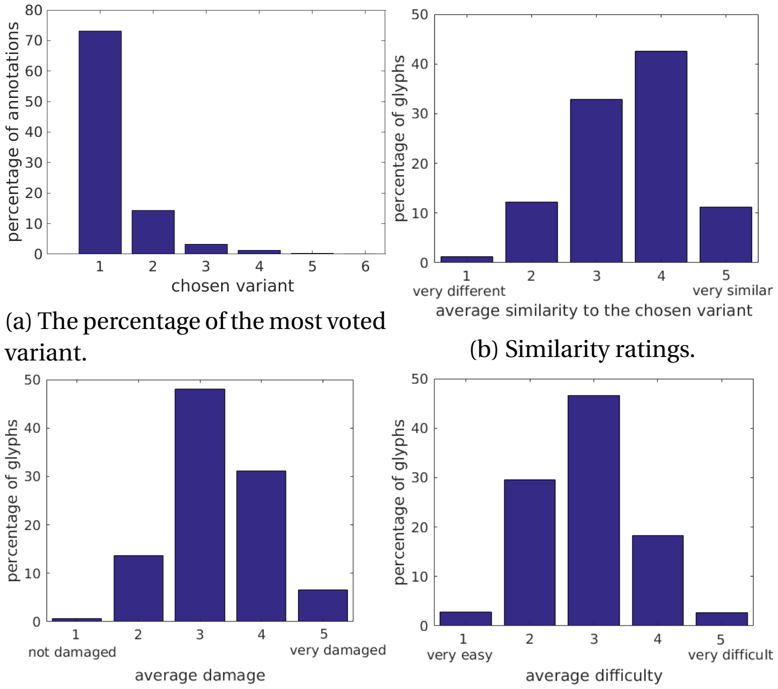
For 23.2% of the annotations, the contributors found that the chosen variant looked different or very different than the glyph they had segmented. On the other hand, only 10.5% of the annotations are marked as "very similar." The reason behind it may be the tendency of workers to be conservative about the visual similarity scale, or indeed due to the visual differences of the glyph regions and the variants.

Task Difficulty and Glyph Damage. For the damage ratings, the general tendency of the contributors (41.9% of the annotations) was to give an average score. However, there are still cases marked as "damaged" or "very damaged" (30.6%), even though we provided glyph cases that are in good condition according to the experts. We believe that workers give relative ratings in the full-scale according to the examples they have previously seen.

In terms of task difficulty, only 16.9% of the annotations have "hard" or "very hard' ratings. This is positive feedback from the crowd about the perception of the task complexity.

Segmentation Analysis. Fig. 4.14b shows the overall f-measure agreement distribution for the large-scale set.

Verification. In this step, we inspected the segmentations to spot problematic cases. For the cases with f-measure agreement above 0.8, there was a small portion of glyphs (318 out of 8229), in which both contributors marked another region as the glyph area. In the cases with low agreement (1991 glyphs with f-measure below 0.8), we checked if the individual



(a) The percentage of the most voted variant.

(b) Similarity ratings.

(c) Damage ratings.

(d) Difficulty ratings.

Figure 4.17 – Distributions of the ratings in the large-scale stage.

segmentations were usable. In these ways, we exploited all the possible useful segmentations.

Minimum IU Threshold. As described in Section 4.4.2, for the first half of the glyphs in the large-scale stage, the minimum intersection-over-union measure between the annotator's segmentation and the ground truth of the test questions was set to 0.7. This threshold was increased to 0.8 for the rest of the glyphs. With this more strict threshold, we observed a 3.8% increase in average median f-measure agreement (from 90.2% to 94.0%) and a 5.7% increase in average mean f-measure agreement (from 82.1% to 87.7%). Overall, the obtained segmentations are of high quality.

Challenging Cases. The difficulty of our task is not uniform across the glyph instances. Fig. 4.18 illustrates some of the cases with high disagreement between segmentations. The main reasons for disagreement are:

- Glyph complexity: Glyphs with a large convex area are easier to segment than concave and discontinuous glyphs, i.e. with many separate parts. In Fig. 4.18c, one contributor selected a concave large glyph (green) somehow resembling the first variant instead of the red target region.

- Confusion due to variants: Some variants are a subset or superset of others (i.e., 2S2), as shown in Fig. 4.18b.

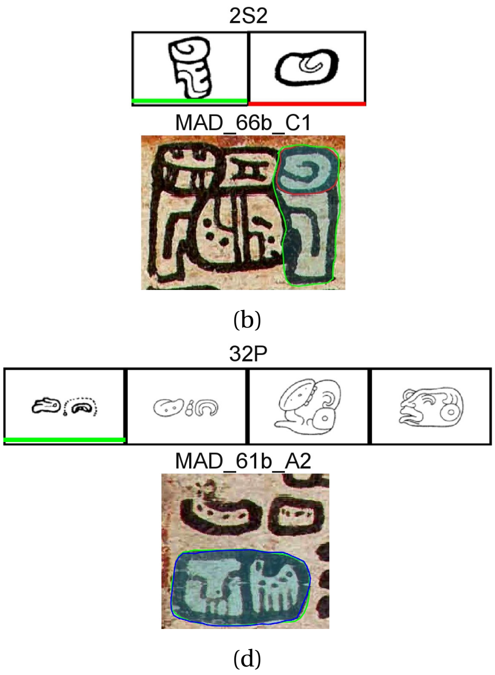


Figure 4.18 – Confused segmentations from the large-scale stage due to (a) similar glyphs in the block, and damaged instances, (b-c) visually-confusing variants, (d) dissimilar glyphs. Red and green colors indicate the markings of the first and the second worker, respectively.

- Dissimilarity between the target region and the variants: We identify three subcases.
  - Target sample not covered by catalog variants. In Fig. 4.18d, the target region is missed by all contributors, and the neighboring glyphs were marked instead.
  - Partial dissimilarity of the glyph. Some glyphs exhibit partial elements different to the variants (Fig. 4.18b).
  - Wrong class annotation. In the process of labeling a glyph with the codes from several catalogs, manual mislabeling is inevitable. We were able to identify few such cases.
- Mismatch of the damage rating between experts and non-experts due to different use of context or visual completeness. In Fig. 4.18a, none of the contributors marked the target region, as the target region is either damaged or lacks partial details.
- Similarity to other glyphs in the block. In Fig. 4.18a, even though the target glyph belongs to class AA1, not HE6, the outline of the neighboring glyph is quite similar to the target region, and the visual difference is subtle.

Outcome. 328 unique contributors participated to the large-scale stage. The average number of glyph annotations per contributor was 66.8 (median 33). This stage produced satisfactory outcomes with two non-experts per sample and minimal manual verification. Overall, we obtained valid segments for 9119 glyphs (together with the ones from the small-scale stage) that are spread over 291 MV categories, with the average f-measure agreement 0.914. Most of these valid segments (8661 out of 9119) belong to the most frequent 150 classes in our dataset. We used these aggregated valid segments in the classification task described in Chapter 5.

4.4.4 Conclusions of Second Task

In this work, we achieved the segmentation of Maya glyphs from three codices (Dresden, Madrid, and Paris) with the help of crowdworkers. The main conclusions are as follows:

- **Task design.** As the data target does not come from everyday objects, guiding non-experts is essential to obtain a satisfactory outcome. From our experience with the task design in the preliminary stage, we observed that a simpler and focused task design (to segment individual glyphs rather than all glyphs in a block) and clear instructions were indispensable.

- **Catalog choice.** From the small-scale stage, we concluded that the variants from the MV catalog matched a higher percentage of the glyph instances compared to the variants from the T catalog. This enabled non-experts to reach a higher consensus on the "closest-looking" variant, and obtain higher agreement (average f-measure). Furthermore, we observed that workers found the task easier with MV variants. These results were to some degree expected as monumental glyphs were the main source of Thompson catalog variants.

- **Non-expert behavior analysis.** We pointed out the main challenges that the crowd faced during the task, such as visual within-class dissimilarities or between-class similarities, and effect of damage. These challenges affect the segmentation outcome. However, they are inherent to the data. That is why our work needed a careful task design, and multi-stage analysis (preliminary, small- and large-scale).

- **Maya codical glyph corpus.** This work generated over 9000 individual glyphs from the three Maya codices along with the corresponding metadata, such as similarity rating of the instances to the MV variants. The dataset will be made publicly available.

4.5 Conclusion

The first crowdsourcing study described in this chapter proved that non-experts perceive the glyph concepts correctly in the non-complex cases (3 glyphs in a block), and when the number of glyphs is given as a means of supervision.

Thanks to the experience acquired by performing the first study, we re-designed the glyph segmentation task for the second study. According to the preliminary stage, we simplified and improved the new task design. The small-scale stage concluded that non-experts were able to generate high-quality segmentations. After the analysis of required number of workers on the small stage, we proceeded to the large-scale stage experiments with 2 workers per glyph. After checking the segmentation agreement, we obtained valid segmentations for over $9K$ glyphs. 8661 glyphs belong to the most frequent 150 classes in our dataset. Note that we noticed some differences in the glyph label annotations of two different experts due to visually-similar glyph samples in a block. According to these annotations, those two experts would have marked different glyph regions in the block. Keeping that this task may be difficult even for experts, we consider that our dataset curated thanks to the non-experts by providing enough guidance is a remarkable achievement. The annotations obtained via the crowdsourcing experiments presented in this chapter enable further studies such as glyph classification and visualizing the diagnostic parts of glyphs. Chapter 5 tackles glyph classification by exploiting the valid segments now available.

5 Learning Shape Representations with CNNs

This chapter focuses on learning shape representations for ancient Maya hieroglyph classification. In document analysis of historical and artistic materials, visual similarity-based recognition is important for helping experts to assess shapes quantitatively and to annotate documents easily. Encoding shapes into discriminative representations is essential for visual similarity tasks. As traditional shape descriptors (e.g. [Lowe, 1999]) have limits for degraded characters in old manuscripts (see Fig. 5.1) or sparse shapes such as sketches, learning robust shape representations from data is of great interest to multimedia research.

In parallel, recent advancements with deep Convolutional Neural Networks (CNNs) for recognition tasks in computer vision, speech, and multimedia have proven the usefulness of these methods [LeCun et al., 2015]. However, applying such techniques when dealing with small amounts of data is challenging. We investigate different strategies to learn such a representation: using a pretrained CNN as a feature extractor, transferring knowledge from a pretrained CNN via fine-tuning, and training a CNN from scratch.

More specifically, since deep representations have been shown to be strong baselines for various visual recognition tasks [Cireşan et al., 2012; Donahue et al., 2014; Sharif Razavian et al., 2014; Simonyan and Zisserman, 2014], assuming that Maya shapes painted on codices share commonalities with hand-drawn sketches and everyday objects in natural images, we

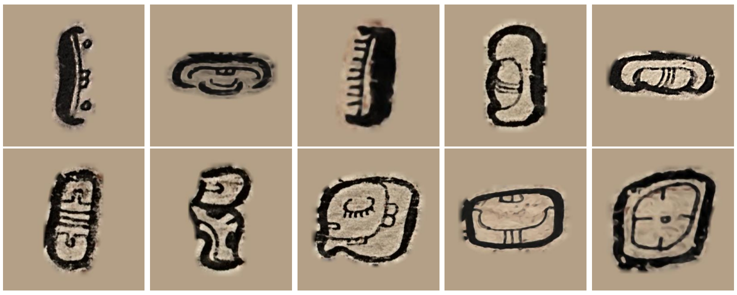


Figure 5.1 – Segmented glyph samples from the 10-class experiment.

have considered the deep representations learned on natural images and on sketches as baseline methods for our task.

We also investigate transferring the knowledge learned on a large-scale dataset to our specific problem by fine-tuning a pretrained CNN as in [Yosinski et al., 2014]. This approach has been shown to be useful in a variety of multimedia applications [Abdulnabi et al., 2015; Cireşan et al., 2012; Tajbakhsh et al., 2016], and is particularly valuable when the data collection is not scalable, either due to the involved cost and time, to ethical and security reasons (i.e. medical data), or simply because of extinct data sources (i.e. scripts from the ancient civilizations).

Usually, training a CNN from scratch requires a large amount of data. However, recent work on batch normalization (BN) [Ioffe and Szegedy, 2015] and dropout regularization [Hinton et al., 2012], along with oversampling with data augmentation (i.e. applying random geometric transformations to the data) enable training a CNN for small- to medium-scale data while reducing data imbalance [Hensman and Masko, 2015] and overfitting issues. Specifically, we study different CNN training approaches for learning visual representations for Maya glyph classification tasks, relying on the data of the three remaining Maya Codices introduced in Chapter 4.

Our contributions are as follows:

1. We evaluate traditional shape descriptors and representations from pretrained networks for the recognition of ancient Maya glyphs;
2. We study different ways to train a CNN for small-scale data, specifically assessing a shallow network over the features from pretrained networks, fine-tuning pretrained networks, and training networks from scratch;
3. We systematically assess a variety of CNN models from old to recent ones, i.e. LeNet, Sketch-a-Net, VGG, ResNet, and that are designed for different purposes (natural image vs. sketch classification).

First, we evaluate the activations from the last block of several pretrained CNNs as shape representations to classify glyphs with a shallow neural network and a linear Support Vector Machine (SVM). This standard transfer learning approach from deep convolutional networks is found promising even in the case of few examples per class (around 80% average accuracy in 150-class case). When VGG-16 network is used, this approach outperforms the traditional shape descriptors by a large margin (around 22% to 37% absolute improvement).

Secondly, we fine-tune the pretrained deep CNNs, and show that in spite of the larger discrepancy between the source (ImageNet) and target (glyph) data, VGG-16 outperforms the dedicated Sketch-a-Net (SaN) that is trained on 250-class sketch data [Eitz et al., 2012a] by 8.9% to 14.8% in average classification accuracy. In particular, we show 2.4 to 14.4% absolute performance improvement by fine-tuning the VGG-16 model, compared to the feature extraction baseline results.

Thirdly, we also train several CNNs from scratch with the glyph data. Interestingly, despite the small amount of data, thanks to batch normalization (BN), data augmentation, and dropout regularization, we achieve outperforming results with a modified, sketch-specific CNN compared to the fine-tuning approach. This point demonstrates that nowadays network training may not necessarily require a large amount of data to be effective.
Overall, our study takes a step towards providing automated tools to scholars in Digital Humanities.

The chapter is organized as follows. Section 5.1 presents the related work. Section 5.2 describes the methodology. Section 5.3 presents the classification results, and Section 5.4 concludes the chapter.

The material presented in this chapter originally appeared in the following papers:

– Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Maya codical glyph segmentation: A crowdsourcing approach. Research Report Idiap-RR-01-2017, Idiap, January 2017c (accepted for IEEE Transactions on Multimedia)

– Gülcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Shape representations for maya codical glyphs: Knowledge-driven or deep? In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, CBMI '17, pages 32:1–32:6, New York, NY, USA, 2017a. ACM. ISBN 978-1-4503-5333-5. doi: 10.1145/3095713.3095746. URL http://bib-ezproxy.epfl.ch:2512/10.1145/3095713.3095746

– Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. How to tell ancient signs apart? Recognizing Maya glyphs with CNNs. Idiap-RR Idiap-Internal-RR-26-2017, Idiap, April 2017b (to be submitted to ACM Journal on Computing and Cultural Heritage (JOCCH))

5.1 Related Work

This section discusses architectures, and common training strategies with CNNs (i.e. analysis of pretrained CNN features, fine-tuning or training deep CNNs from scratch); previous glyph and sketch recognition studies; and visualization and interpretation of the learned representations.

CNN architectures. For handwritten digit classification, LeCun et al. [1998a] proposed a sequential network (LeNet) with three convolutional layers followed by a nonlinearity activation (e.g. sigmoid), and subsampling (e.g. max pooling). Krizhevsky et al. [2012] built upon the LeNet architecture, and proposed the AlexNet that has five convolutional layers with non-saturating rectified linear unit (ReLU) activations for object classification on ImageNet data

[Deng et al., 2009]. Vanishing/exploding gradients during error backpropagation through this deep network were handled by the ReLU activations. Additionally, the dropout strategy [Hinton et al., 2012] helped to prevent overfitting during training of the AlexNet. VGG networks illustrated the limits of deep sequential networks without any special design [Simonyan and Zisserman, 2014]. Simonyan and Zisserman [2014] showed that such a network with 16 layers (VGG-16) outperforms the AlexNet.

More recently, graph-based architectures have emerged in CNN design such as GoogleNet with inception modules [Szegedy et al., 2015, 2016], and residual networks (ResNets) with identity mapping connections [He et al., 2016]. One important commonality of these models is heavy-usage of Batch Normalization (BN) [Ioffe and Szegedy, 2015] which enables to train very deep networks in a considerably short amount of time with improved performance, since BN reduces covariate shift in the data during training.

Transfer learning and training with CNNs. Motivated by the common visual structures learned by CNNs in the first layers, several transfer learning approaches reutilized and analyzed the effectiveness of pretrained CNN representations on different datasets [Cireşan et al., 2012; Donahue et al., 2014; Sharif Razavian et al., 2014; Simonyan and Zisserman, 2014]. The penultimate activations of a CNN, specifically AlexNet [Donahue et al., 2014; Sharif Razavian et al., 2014] and VGG [Simonyan and Zisserman, 2014] trained on ImageNet data, are shown as strong baselines for visual recognition tasks. Similarly, for character recognition tasks, Cireşan et. al. indicated that existing pretrained networks can be utilized as feature extractors [Cireşan et al., 2012].

Alternatively, these pretrained networks can be fine-tuned for new tasks, and this fine-tuning helps the training start from a more relevant point and results in improved performance and faster training compared to random initialization [Cireşan et al., 2012]. Furthermore, Yosinski et. al. showed that fine-tuning the last convolutional layers helps the network to learn representations that are more specific to the target dataset [Yosinski et al., 2014]. Authors discuss that fine-tuning more number of layers (from last fully-connected layer towards input layer) might be essential as the nature of the target dataset becomes more different than the initial source dataset used for pretraining.

As our glyph data has its own particular characteristics, our third strategy (training a CNN from scratch) can be considered as an alternative to fine-tuning all the layers. Note that we utilize batch normalization (BN) layers in the case of full CNN training. Hence, even though we start with a Glorot initilization of the weights [Glorot and Bengio, 2010], BN compensates for not utilizing pretraining.

Glyph and sketch recognition. For Maya glyph recognition, several shape representations have built upon traditional knowledge-driven descriptors [Hu et al., 2015; Roman-Rangel et al., 2011a]. These representations are based on bag-of-words (BoW) that output the frequency histograms of local shape descriptors. As shown in a similar study on Egyptian glyphs [Franken and van Gemert, 2013], HOOSC [Roman-Rangel et al., 2011a] is a competitive candidate among other traditional shape descriptors.

On the other hand, for shape encoding with neural networks, in Chapter 3, we discussed a single-layer Sparse Autoencoder (SA), which encodes the same local regions as HOOSC. This local SA representation is competitive for 10-class monumental glyph classification task. However, this shallow representation is not representative enough for other tasks, i.e. sketch classification task proposed in [Eitz et al., 2012a]. Due to the scarcity of the strokes in thin sketch drawings (and the high variety of the drawings), the BoW frequencies of the simple edge encodings in the shallow sparse encoder were reported to be harder to capture than relatively-thicker glyph strokes. We concluded Chapter 3 by discussing that deeper convolutional networks trained with more data might capture more complex and more discriminative shape representations for this case. Complementary to this finding, the "Sketch-a-Net" [Yu et al., 2015] has illustrated that a modified version of the AlexNet (in multiple scales and multiple temporal channels) beats human performance on the 250-class sketch dataset of [Eitz et al., 2012a]. This model has fewer feature maps, yet larger first layer convolution kernels compared to the AlexNet, which is designed for natural images.

In the context of Maya glyph-block retrieval, in our joint study, Roman-Rangel et al. [2016] showed that the middle-layer activations (conv5) of VGG outperform the last-layer activations (fc-7), and the bag-of-words representation of a traditional shape descriptor (HOOSC). This is a motivating point for learning the representations for Maya glyphs, and taking advantage from existing pretrained networks.

5.2 Methodology

To evaluate the shape representations for glyph recognition tasks, we considered

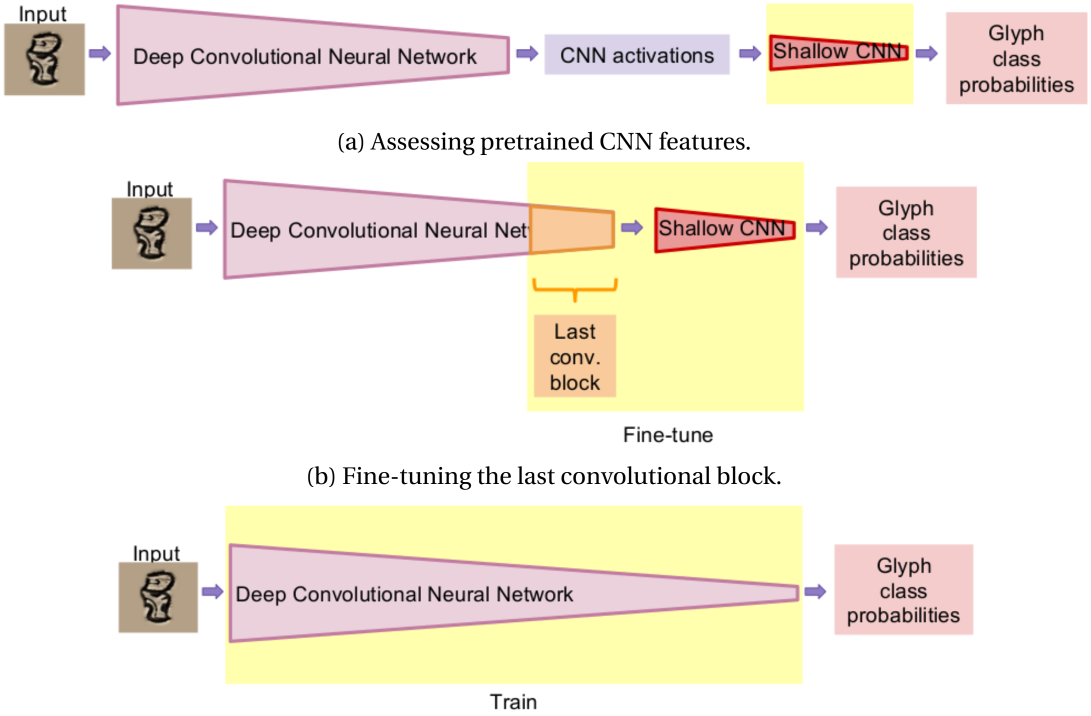
1. two traditional shape descriptors, i.e. the bag-of-words representation of a local shape descriptor (HOOSC) [Roman-Rangel et al., 2011a], and a multi-level HOG [Dalal and Triggs, 2005b];

2. assessing features learned by a pretrained network [Donahue et al., 2014; Sharif Razavian et al., 2014];

3. transferring knowledge from an existing network through fine-tuning; and

4. full network training.

Fig.5.2 illustrates our data-driven approaches. The first and second data-driven approaches are good alternatives to the last one when dealing with the challenge of small amount of data, as in our case. We describe each of these methods below.


Input
Deep Convolutional Neural Network
CNN activations
Shallow CNN
Glyph class probabilities
Input
Deep Convolutional Neural Net
Shallow CNN
Glyph class probabilities
Last conv. block
Fine-tune
Input
Deep Convolutional Neural Network
Glyph class probabilities
Train

(a) Assessing pretrained CNN features.

(b) Fine-tuning the last convolutional block.

(c) Full network training.

Figure 5.2 – Three data-driven methods for supervised glyph classification. In each method, only the highlighted part of a CNN model was trained.

5.2.1 Traditional Shape Descriptors

For the bag-of-words on the HOOSC descriptors, we followed the same pipeline in Section 3.3.2 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 [Otsu, 1975] (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 we 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. 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 pivots or more ($0.1 * N_{skeletonpoints}$) from each glyph skeleton if possible, otherwise we used all the skeleton points as pivots.

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.

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. The final representation HOOSC-BoW has 4000 dimensions.

Multi-Level HOG Descriptor Extraction. We concatenated the histogram of orientation features at two-levels. We computed the HOG with 13×13 and 24×24 pixels cell sizes and 4 blocks in each cell with 9 orientations. Since our images have 224 pixel image size, we ended up with $16 \times 16 + 8 \times 8 = 320$ cells, and $320 * 4 * 9 = 11520$ feature dimension for each image.

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 equal to 1, and then scaled the BoW vectors with a constant scalar to match the maximum activation value of the pretrained CNN features. A similar normalization is applied to the HOG features.

Classification. The HOOSC-BoW and multi-level HOG features are used as input to a shallow neural network (Fig. 5.3) with two fully-connected (FC) layers. The first FC layer has 1024 filters. We applied ReLU activation between two FC layers as well as batch normalization [Ioffe

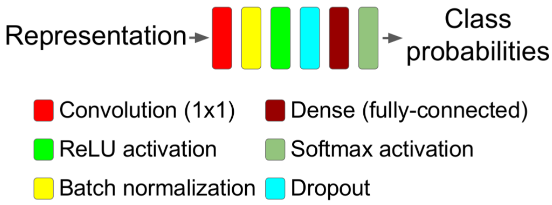


Figure 5.3 – Shallow CNN model used in Section 5.2.2.

and Szegedy, 2015], and dropout [Hinton et al., 2012] method with 0.5 rate. The final class probabilities are determined by the softmax activation at the end. Additionally, we assessed the representations with a standard linear support vector machine (SVM) as well.

5.2.2 Pretrained CNN Features

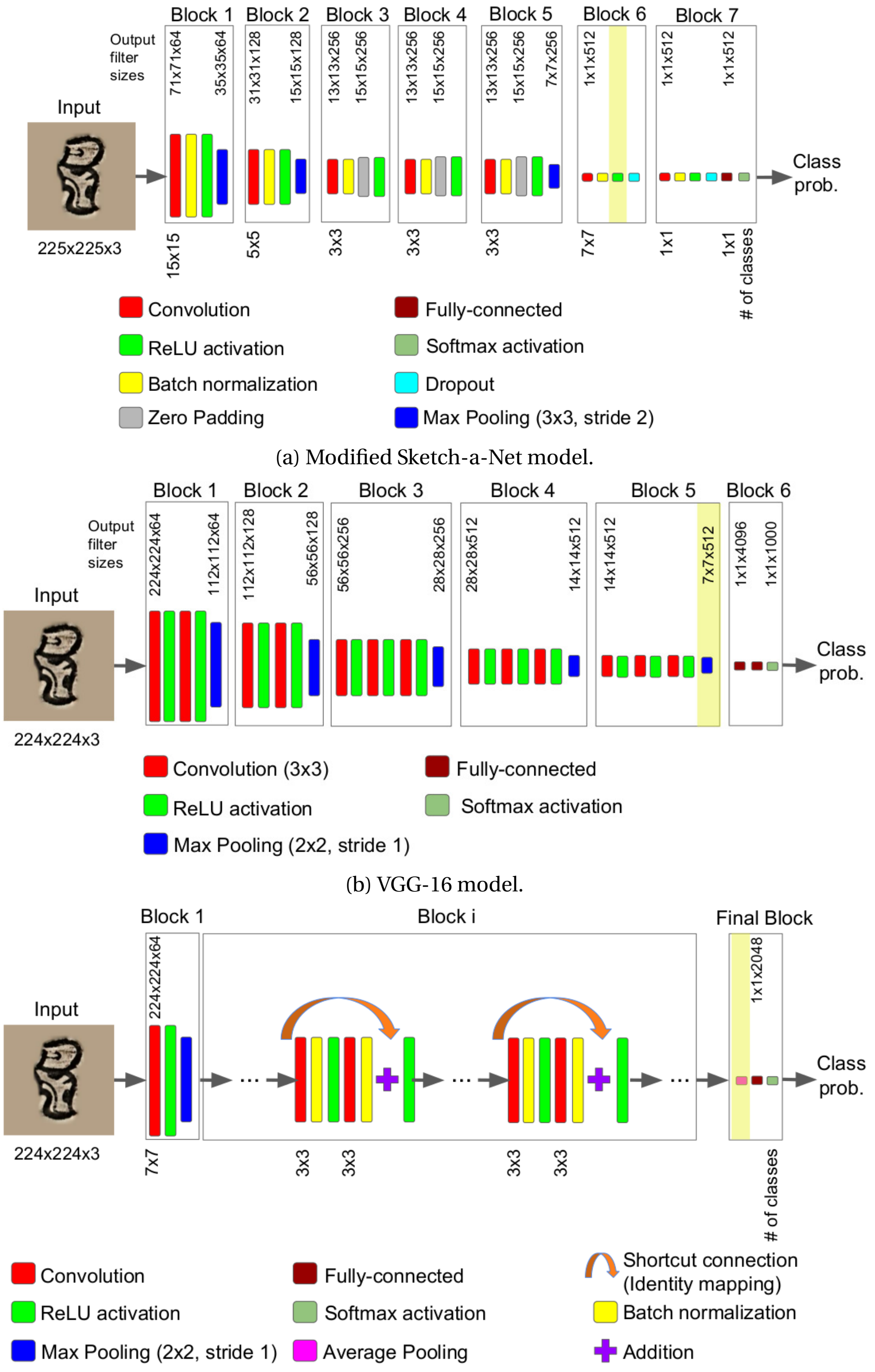
CNNs pretrained on large-scale datasets, i.e. ImageNet, are used as feature extractors by feedforwarding the image of interest, and gathering the activations at different layers of the network [Donahue et al., 2014; Razavian et al., 2016; Sharif Razavian et al., 2014; Tolias et al., 2015; Yosinski et al., 2014; Zheng et al., 2017]. The penultimate activations before softmax classifier have been reported as good baselines for transferring knowledge in several vision tasks [Donahue et al., 2014; Sharif Razavian et al., 2014]. Furthermore, 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) [Yosinski et al., 2014].

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. To assess these representations, the same classifiers were applied as noted in Section 5.2.1.

Method. Given a pretrained network, and following [Hoffman et al., 2013], we use the output of the last convolutional or residual block from all glyph images in the training set as training features of a one-layer convolutional network (denoted as *B*). As depicted in Fig. 5.3, this shallow network is composed of 1x1 convolution layer with 512 feature maps followed by batch normalization, ReLU nonlinearity, dropout, and a softmax classifier. Compared to the alternative approach of fine-tuning the fully-connected (FC) block of the pretrained network with our own data, this is much less costly.

Considered Networks. We considered the VGG-16 network Simonyan and Zisserman [2014] and ResNet-50 He et al. [2016] pretrained on ImageNet dataset, and the Sketch-a-Net Yu et al. [2015] pretrained on 250-class binary sketch images Eitz et al. [2012a].

VGG-16 is a 16-layer CNN model, shown to be competitive on the ImageNet dataset before the inception module and residual connections were introduced. We also experimented with the state-of-the-art residual network that uses BN layers and residual connections. On



(a) Modified Sketch-a-Net model.

(b) VGG-16 model.

(c) ResNet-50 model.

Figure 5.4 – Considered pretrained CNN models in Section 5.2.2. After a single forward-pass of a glyph image through a network, activations from a highlighted layer are used as pretrained CNN features.

the other hand, Sketch-a-Net (SaN) is an altered version of AlexNet for handling the sparse strokes with larger convolution kernels in the first layers. As there are fewer feature maps in the convolutional layers, this network has around $8.5M$ parameters compared to the $60M$ parameters of the original AlexNet.

For the VGG-16 and R50 models, we utilized existing pretrained models. However, due to technical incompatibilities, we retrained the single-scale, single-channel version of SaN from scratch, by adding batch normalization (BN) layer Ioffe and Szegedy [2015] after each convolutional and dense layer (see Fig. 5.4a). Furthermore, we utilized the Adam adaptive gradient update scheme [Kingma and Ba, 2014] , which was shown by Kingma et. al. to outperform other adaptive SGD-based optimizers, as the gradients get sparser at the end of the optimization. The modified SaN 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. Similarly, we retrained another SaN with the fake-colored sketch images (background filled with the same RGB values that are used to populate our glyph dataset).

With these configurations, we obtained 71.4% average test accuracy for 9750 colored sketches from the 250 classes (a random $\frac{1}{6}$th split of the 60K data, as another $\frac{1}{6}$th is used for validation and the rest $\frac{2}{3}$th is used for training).

The feature dimensions extracted from the pretrained nets are 2048 for the R50 (output of the global average pooling layer), $512 \times 7 \times 7 = 25088$ for the VGG-16 (output of the 5^{th} block), and 512 for the SaN (output of the 6^{th} block).

5.2.3 Network Adaptation

This method consists in jointly fine-tuning the last block of the pretrained network along with the weights of the shallow network *B*. More precisely, we replace the dense (fully-connected) layer at the end of the original network with the shallow network depicted in Fig. 5.3. After freezing the weights in the original network up to the last block, we train the whole “stitched” model altogether with the glyph images. Specifically, the 6^{th} block in the Sketch-a-Net (SaN) pretrained model (see Fig. 5.4a), the 5^{th} convolutional block in the VGG-16, and the 5^{th} residual block in ResNet50 (R50) are fine-tuned.

For the shallow network at the top, we start from the weights trained as in Fig. 5.3, allowing the optimizer to start from a more relevant initialization than a random one. Additionally, the optimizer learning rate is set smaller, to prevent large gradient updates and disrupt the pretrained weights within the last convolutional block.

This fine-tuned network is denoted as *F*.

5.2.4 CNN Training from Scratch

We also investigated training of CNNs from scratch. Given our amount of data, networks with fewer parameters are preferable. As first choice, we utilized the classic LeNet model, however with ReLU activations, additional batch normalization after convolutional layers, and dropout strategy. This modified LeNet model has $44M$ parameters to learn. Secondly, the sketch-specific SaN with additional BN layers was trained. Our third choice is the recent residual networks. Considering the number of parameters, we decided to train two versions of the ResNet with 18- and 50-layers ($11M$ and $25M$ parameters respectively).

Another commonly-adopted option would be to train an inception network. Although the inception-v4 model has similar number of parameters, it has outperformed the R50 on ImageNet data for top-1 accuracy [Szegedy et al., 2017]. However, Canzani et. al. have shown that the contribution of number of parameters to top-1 accuracy is higher for the R18 and R50 models compared to the inception-v4 model [Canziani et al., 2016]. This implies that the information density stored in the neurons of these residual networks are higher than the inception-v4 model. This accuracy vs. parameter analysis is especially important for training a CNN from scratch with small- to medium-scale data. Therefore, we omit the inception-v4 model comparison here.

5.3 Settings and Results

5.3.1 Experimental Settings

Data Preparation. To assess the difficulty of our glyph dataset that was introduced in Chapter 4, we experimented with different number of classes (the most frequent ones). We have 11 classes with more than 200 such glyphs, whereas, at the other end of the spectrum, 52 classes have just one such glyph. Table 5.1 shows the number of glyphs for each experimental setting (the maximum number is 384).

We applied some preprocessing steps to facilitate the training process. The large variance in the existence and composition of glyph parts motivated us to simplify the classification task by eliminating the noisy background, that might contain parts of other glyphs in the block, around the target glyph region. Thus, instead of using directly a bounding box around a glyph segment on an original codices page, we generated a square bounding box around the glyph segment with different background colors. Since we studied training CNN models that contain fully-connected layers (i.e. not fully-convolutional), we standardized the dimensions of input images to be square and same rather than using rectangular images with arbitrary shapes. Note that we kept red-green-blue (RGB) color channels of the original images in order not to lose information during binarization.

In detail, for each glyph, to obtain a square crop centered on the aggregated segmentation mask, we applied the following steps.

Table 5.1 – Number of glyphs for the classification tasks.

		Number of classes				
		10	**30**	**50**	**100**	**150**
Number of samples	**min**	211	83	50	20	5
	mean	255.7	176.16	132.66	81.19	57.74
	median	235	173	101	50	27
	total	2557	5285	6633	8119	8661

1. *Dilation:* We dilated the aggregated mask in case of segmentation not covering all boundary pixels. We set the dilation dynamically as 1/32 of the long edge size of the bounding box.

2. *Color filling:* We sampled 3 RGB colors from background areas of the codices. Additionally, we computed a dynamic RGB value from each block image as $0.65 * threshold$ using Otsu's method [Otsu, 1975]. In the need of padding, we filled the areas with these RGB values. Note that this step quadruples the number of samples per class.

3. *Padding:* For convenience during convolution, we applied padding around all the edges for 1/6 of the long edge size of the dilated aggregated mask. Then, we padded the short edge to make the final crop square-sized.

4. *Scaling:* We scaled all processed square crops to 224 × 224 pixels.

Note that similar to the original papers, the image width is set to 224 pixels when using the VGG-16, ResNet, and LeNet models; whereas, for the SaN case, it is set to 225 pixels.

After these preprocessing steps, we shuffled and divided each set of glyphs to training (60%), validation (20%), and test sets (20%). We repeated this splitting five times to generate 5 data folds. We report the average accuracies among the 5-folds.

Sampling Strategy. In this chapter, *sampling* refers to selecting a predetermined number of data samples. In this context, we use the term of *original sampling*, when we employ all the available samples as shown in the last row of Table 5.1. Furthermore, we use the term of *undersampling*, when we use the dataset partially by selecting a certain number of samples per class. To the contrary, the term of *oversampling* is used when we overpopulate the dataset by replicating some randomly-chosen samples with small perturbations so that each class would have the same number of samples.

To handle the data imbalance issue among the categories, we considered undersampling and oversampling as alternative strategies to the original sampling. For undersampling, we randomly picked the same number of samples per class in each experiment. Based on the minimum numbers in Table 5.1, we chose 200, 80, 48, 20, and 5 samples, respectively.

For oversampling, we applied random geometric data augmentation, comprising rotation (within $[-15, 15]$ degrees), vertical and horizontal translation ($\pm 0.1\times$ image width), and zooming (scale within $[0.8, 1.2]$). We oversampled the existing examples such that each class had 1000 training, 300 validation, and 300 testing samples. Therefore these oversampled sets were a mix of original data and synthetic data. Note that these applied geometric transformations to populate the synthetic data do not alter the nature of the data. On the contrary, these transformations imitate the possible scenarios during the acquisition of the original codices page images, i.e. scanning the pages with a translation offset, with a small angular rotation, or with a different degree of zooming. Thus, we consider these populated synthetic images can be treated in the same way as the original images.

Tasks and performance measures. We considered four tasks: 10-, 50-, 100-, and 150-class glyph classification. Note that the samples in the 10, 50, and 100 classes were chosen from the 150-class set according to the criteria that they had the most number of samples per class. We reported the average test accuracy (top-1) along with the top-5 accuracy (i.e. true class is in the top 5 predicted classes).

Note that in Table 5.2 and 5.3, we reported the average accuracies on the *original* test sets, whereas in Table 5.4, we reported the average accuracies on the *oversampled* test sets. By reporting on the oversampled test sets, we aimed to avoid the performances of the frequent classes to affect the overall performance.

Training Settings. For training the shallow network over pretrained features and training the CNNs from scratch, Adam optimizer was used with the learning rate 10^{-5} and 10^{-4} respectively. For fine-tuning, Stochastic Gradient Descent (SGD) with momentum (0.9) was used. The learning rate was set to 10^{-4}, and it was reduced with a factor of 0.2 when the validation loss was not decreasing for 10 epochs. We applied model check-pointing (keeping track of the parameters that result in highest validation accuracy during the optimization) during all the training cases. For pretrained network training and fine-tuning, the maximum number of epochs was set to 500 empirically, whereas for training from scratch, we followed an early-stopping approach with a patience factor of 20 epochs, i.e. terminating training if validation loss had not decreased for 20 epochs.

We performed each of our experiments on a single core on either NVIDIA Tesla K40m or K80 GPU hardware. We observed that training Sketch-a-Net or ResNet-50 from scratch with oversampled glyph images converges within around 40 hours with an average 0.5 (with ResNet50 for 50-class case) to 1.5 hour per epoch (with Sketch-a-Net for 150-class case).

5.3.2 Classification Results

The classification results are presented and discussed under two main points below.

Comparing Traditional Descriptors vs. CNN Representations. Table 5.2 shows the average accuracies among 5-fold experiments with original sampling in different settings. As the number of classes increases and the number of samples per class decreases, the classification

Table 5.2 – Average classification accuracies (and standard deviations among 5-folds) of the original test sets with a linear SVM (S) and the shallow CNN (N) in Fig.5.3.

	Original Sampling							
	Number of classes							
	10		**50**		**100**		**150**	
Model	**S**	**N**	**S**	**N**	**S**	**N**	**S**	**N**
HOOSC	70.1 ± 2.4	69.8 ± 1.8	49.5 ± 1.0	50.1 ± 0.9	44.0 ± 0.7	43.1 ± 0.6	39.7 ± 0.7	40.3 ± 0.9
HOG	67.2 ± 2.8	71.1 ± 1.0	46.0 ± 0.6	50.3 ± 1.4	41.8 ± 1.2	44.5 ± 0.7	39.2 ± 0.7	41.4 ± 0.4
SaN_B	81.6 ± 1.2	85.7 ± 0.8	63.5 ± 1.2	71.6 ± 1.7	58.2 ± 1.9	66.0 ± 1.4	56.1 ± 0.6	63.4 ± 1.0
SaN_RGB	84.4 ± 1.6	88.6 ± 1.5	70.2 ± 1.0	77.0 ± 0.8	65.2 ± 1.3	73.0 ± 1.4	62.5 ± 0.9	70.1 ± 0.7
VGG16	**92.0 ± 0.5**	**91.8 ± 0.7**	**86.6 ± 1.0**	**84.2 ± 0.7**	**82.6 ± 0.9**	**82.3 ± 0.7**	**80.0 ± 0.6**	**79.2 ± 0.6**
R50	75.7 ± 2.2	81.7 ± 1.2	51.8 ± 4.9	68.1 ± 1.1	46.0 ± 5.5	63.2 ± 0.6	41.5 ± 1.5	59.5 ± 1.2

Table 5.3 – Average accuracies on the original test sets for pretrained features, when the shallow CNN networks were trained on the undersampled vs. oversampled sets.

		Number of classes			
	Model	**10**	**50**	**100**	**150**
Undersampling	SaN_RGB	87.9 ± 1.8	67.6 ± 0.8	54.6 ± 1.4	29.1 ± 1.9
	VGG16	91.3 ± 0.9	78.0 ± 0.6	64.1 ± 0.5	35.2 ± 0.8
	R50	79.4 ± 1.6	51.4 ± 5.6	35.9 ± 1.1	16.5 ± 1.3
Oversampling	SaN_RGB	95.6 ± 0.9	91.5 ± 0.2	90.0 ± 0.6	71.4 ± 0.8
	VGG16	97.0 ± 0.9	95.0 ± 0.4	93.6 ± 0.8	80.6 ± 0.4
	R50	93.5 ± 1.5	88.2 ± 1.0	86.1 ± 0.6	62.0 ± 0.6

Table 5.4 – Average top-1 (T-1) and top-5 (T-5) accuracies of the oversampled test sets. Models: L: LeNet, SaN: Sketch-a-Net, R: ResNet, VGG-16. Conditions: B: pre-trained, F: fine-tunning, S: learned from scratch. Best performances in B, F, and S are in bold.

	Number of classes							
	10		**50**		**100**		**150**	
Model	**T-1**	**T-5**	**T-1**	**T-5**	**T-1**	**T-5**	**T-1**	**T-5**
SaN-B	81.5	98.0	65.2	85.8	50.5	71.2	42.3	62.7
VGG-B	**87.9**	**99.1**	**77.1**	**91.9**	**62.4**	**81.3**	**52.8**	**74.0**
R50-B	77.8	96.4	54.3	79.0	50.3	75.0	31.8	52.8
SaN-F	81.0	98.5	70.9	90.1	58.7	80.3	50.2	72.7
VGG-F	**89.9**	98.9	**85.4**	**95.9**	**73.5**	**89.2**	**64.5**	**83.5**
R50-F	87.0	**99.3**	79.4	93.9	70.5	87.7	60.4	78.9
L-S	81.4	98.2	70.1	89.4	52.2	74.9	46.0	68.0
SaN-S	**91.0**	**99.8**	**87.8**	95.9	75.1	90.1	**70.3**	85.4
R18-S	88.0	99.6	87.3	**96.9**	78.6	94.3	68.6	**85.5**
R50-S	89.1	99.6	85.1	96.6	**79.2**	**95.3**	67.4	84.6

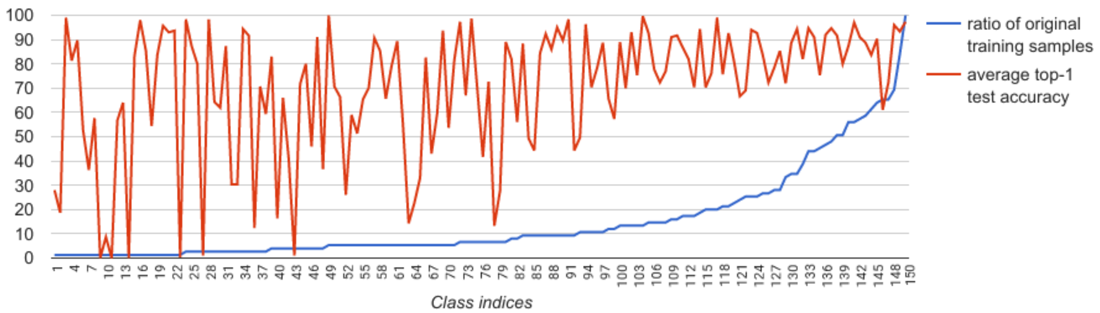


Figure 5.5 – Ratio of original training samples vs. average top-1 class accuracies for the SaN-S model for the 150 classes. Despite the performance fluctuations among classes, top 50 most frequent glyph classes exhibit a high performance trend, showing the importance of original training data size.

problem becomes more challenging. With 200 glyphs per class in the 10-class experiment, we obtained 91.8% average accuracy with the VGG-16 pretrained features. For the 150-class case, we obtained 79.2% accuracy (random guess would be 0.66%). Table 5.2 confirms the competitiveness of the pretrained CNN features, that are learned from large-scale datasets, compared to traditional shape descriptors. Another point is using colored images rather than binarized glyph images resulted in 3 to 7% accuracy improvement in case of the pretrained Sketch-a-Net features. Among the pretrained network features, the VGG-16 activations provide the best results.

Table 5.3 points out that oversampling during training helps all the models and improve over undersampling with a large margin. Thanks to the random geometric transformations applied to oversampling the data, the learned visual representations were more robust to translation, rotation, changes in the scale and partial matching (thanks to zooming). These accuracy improvements were reported based on the original test data (without any transformations). During our experiments, we observed that oversampling also helps on improving the performance on the populated/oversampled test data.

These results from both tables show the challenges and complexity of our dataset.

Assessing CNN Representations. We experimented with three main training strategies, and with a range of different models as follows:

1. **Pretrained Features.** We train a shallow neural network with the pretrained features (SaN-B, VGG-B, or R50-B),
2. **Fine-tuning.** We fine-tune the last block of Sketch-a-Net with batch normalization that is pretrained on RGB populated sketch images (SAN-F); VGG-16 network pretrained on ImageNet (VGG-F); ResNet-50 pretrained on ImageNet (R50-F); or
3. **Learning from scratch.** We train from scratch LeNet with batch normalization (L-S); Sketch-a-Net with batch normalization (SaN-S); ResNet-18 (R18-S); and ResNet-50

(R50-S).

Table 5.4 presents the results obtained with these methods. Except for the 10-class case of the SaN models, we observe 2.0 to 28.6% absolute improvement in top-1 average accuracies of the fine-tuned *F* models compared to the corresponding pretrained *B* models. Especially, R50 benefits highly from fine-tuning. The most notable point in Table 5.4 is the superior performances of the SaN and R50 models when trained from scratch. These models perform consistently better than their corresponding fine-tuned models.

When training from scratch, with classic sequential networks, the SaN model that is deeper and has fewer parameters than LeNet, always performs better (10.4 to 24.3%). However, performances of the residual networks are similar, and the marginal differences do not allow us to conclude that the shallower network with fewer parameters always perform better with this special network design (as the identity mapping residual connections help the network behave as if it had dynamic depth).

Fig. 5.5 shows the individual class accuracies (average test top-1) compared to the ratio of the training samples for 150-class Sketch-a-Net trained from scratch. This plot demonstrates a lot of fluctuations in the class accuracies. We hypothesize that this is due to the nature of the data, as some classes have small within-class variation and large between-class variation, so that they can be classified well even with small amount of training samples. The maximum number of training samples is 900. In spite of the fluctuations, we can observe the trend of increasing accuracy, especially when the number of training samples is more than 60 (that is 6.6% original samples among the populated 900 samples. This trend is more visible from the class index 80 and on).

Another question is how the performance on the existing classes get affected when a model is trained with additional classes and in general more data. We inspected the class accuracies of the 50 classes from the 50-class and 150-class SaN-S models. We observed that the performance for 9 classes dropped 10% or more, however the performance for 14 classes improved when trained within 150 classes. As expected, in the 150-class set, the classes in the 50-class set get more competitors that share local features. In the discussion that follows, we refer to signs using the Macri-Looper catalog naming system [Macri and Looper, 2003]. For instance, due to the inclusion of other numerical signs, the performance of class 003 (literally 3 as three horizontal thick dots) drops. Similarly, sign XE1's performance drops due to more similar-looking classes coming into play such as signs XE3 and XE7 (all have dots and vertical parallel lines inside a square thick contour). Another notable difference is the 38% increase in the performance of sign 1SD whereas its competitor sign 1SC's performance drops by 17%. This might be due to the inclusion of more classes, which forces the network to spot more subtle differences, or due to the inclusion of more head-signs (increasing in number from 4 to 29) so that the network becomes better at spotting "eyes" and distinguishing them from any other random circle (sign 1SD looks like a profile head sign with a prominent eye and mouth, whereas sign 1SC has two components: a head-like circle with three small inner circles, and a body with inner details similar to 1SD).

From our experiments, we summarize the following trends:

1. VGG-16 seems to have more robust pretrained representations than the ResNet-50 features.
2. Fine-tuning improves the results compared to the pretrained feature classification baseline.
3. Oversampling is essential for handling imbalanced small-scale glyph datasets.
4. Batch normalization and dropout enable training a CNN from scratch with medium-scale oversampled data, and outperform fine-tuning results with a sketch-specific network.

5.4 Conclusion

In this chapter, we studied two traditional shape descriptors and three training approaches with CNNs for our challenging Maya codical glyph dataset. Specifically, we assessed HOG and HOOSC descriptors, representations learned in several existing pretrained networks, i.e. Sketch-a-Net, VGG-16, and ResNet-50, fine-tuning the last blocks of these pretrained networks, and training the sequential and residual CNN variants from scratch (LeNet, SaN, R18, and R50).

We showed that pretrained CNN representations outperform traditional descriptors by a large margin. Furthermore, transfer learning via fine-tuning of the last convolutional layer improved the classification performances considerably, compared to evaluating the pretrained representations directly. We observed that VGG-16 pretrained network is more robust than the recent R50 for assessing the data with different nature (as is the case of glyphs). That said, training a sequential sketch-specific network with few parameters from scratch with batch normalization, balanced oversampling, and dropout regularization outperformed the other training strategies and the recent residual models. Note that this model achieved over 70% average top-1, and over 85% average top-5 accuracy in the 150-class case. This finding is quite promising for all the other visual shape recognition tasks with limited amount of data.

6 Glyph Visualization

Signs following ancient Mesoamerican representational conventions end up being classified according to their appearance, which can lead to potential confusions as the iconic origin of many signs and their transformations through time are not fully understood. For instance, a sign thought to fall within the category of "body-part" can later be proven to actually correspond to a vegetable element (a different semantic domain).

Fig. 2.3 illustrates the challenges to analyze Maya glyphs visually. We posit that adding functionality that take context (i.e., characteristics of the data) and part-whole relations (i.e. highlighting diagnostic parts) into account would bring guidance during decipherment tasks. The tools we envision are different from existing page-by-page visualization systems [Vail and Hernandez, 2013]. They could also be more engaging for users (i.e. visitors in museums), and offer promising perspectives for scholars.

This motivates the study of data visualization. In this chapter, we first built a prototype for visualization of glyphs based on visual features. We introduce an approach to analyze Maya glyphs combining (1) a visual shape representation, and (2) a non-linear method to visualize high-dimensional data. For the first component, as a knowledge-driven representation, we use a bag-of-words representation [Sivic and Zisserman, 2003a] of the local Histogram of Orientation Shape Context (HOOSC) descriptor [Roman-Rangel et al., 2011a,b, 2013]. The HOOSC descriptor has similarities to other descriptors in the visual recognition literature [Belongie et al., 2002; Dalal and Triggs, 2005a; Lowe, 2004], but is adapted to shape analysis, as Franken and van Gemert [2013] showcased in a comparative analysis for Egyptian glyph recognition. Moreover, as data-driven visual representations, we use representations from a deep CNN, namely AlexNet [Krizhevsky et al., 2012] that is pretrained on Imagenet dataset. We obtain the corresponding representations of our glyph data by feedforwarding our images through the pretrained network.

For the second component, we use the t-distributed Stochastic Neighborhood Embedding (t-SNE) [Van der Maaten and Hinton, 2008], which is a dimensionality reduction method from the machine learning literature that has value for Digital Humanities (DH), as it can highlight the structure of high-dimensional data, i.e. multiple viewpoints among samples. As analysis of DH data is often based on attributes like authorship, produced time, and place, observing these variations as smooth transitions with t-SNE becomes a relevant feature.

We show that the proposed visualization methodology is useful to analyze the extent of spatial support used in the shape descriptor and to reveal new connections in the corpus through inspection of glyphs from stone monuments and glyph variants from catalog sources. In particular, we hope that the presentation of our use of t-SNE can motivate further work in DH for other related problems.

As a second contribution, we study methods for visualization of glyph representations from the previous chapters. We evaluate the discriminative parts of ancient glyphs learned by the network using CNN-derived visual explanations. This can provide an interpretability capacity, which is important for domain experts. We visualize the discriminative parts of glyphs via gradient backpropagation [Simonyan et al., 2013] and Grad-CAM [Selvaraju et al., 2016], and show that the trained model has interpretability potential, as the discriminative parts of glyphs overlap with the expert descriptions in a 5-glyph case study. Additionally, we discuss the potential of the Grad-CAM method in glyph localization in a cluttered setting, i.e. glyph-blocks.

The contributions presented in this chapter originally appeared in the following papers:

- Gulcan Can, Jean-Marc Odobez, Carlos Pallan Gayol, and Daniel Gatica-Perez. Ancient Maya writings as high-dimensional data: a visualization approach. In *Digital Humanities*, 2016b.
- Edgar Roman-Rangel, Gulcan Can, Stephane Marchand-Maillet, Rui Hu, Carlos Pallan Gayol, Guido Krempel, Jakub Spotak, Jean-Marc Odobez, and Daniel Gatica-Perez. Transferring neural representations for low-dimensional indexing of Maya hieroglyphic art. In *ECCV Workshop on Computer Vision for Art Analysis*, October 2016.
- Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Maya codical glyph segmentation: A crowdsourcing approach. Research Report Idiap-RR-01-2017, Idiap, January 2017c (accepted for IEEE Transactions on Multimedia).
- Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. How to tell ancient signs apart? Recognizing Maya glyphs with CNNs. Idiap-RR Idiap-Internal-RR-26-2017, Idiap, April 2017b (under submission).

6.1 Related Work

Visualizing representations in low dimensions. Hinton and Roweis [2002] proposed Stochastic Neighborhood Embedding (SNE) as a non-linear dimensionality reduction method. It relates the Euclidean distances of samples in high-dimensional space to the conditional probability for each point selecting one of the neighbors. Van der Maaten and Hinton [2008] proposed to model these distributions as heavy-tailed t-distributions that resulted in the t-distributed Stochastic Neighborhood Embedding (t-SNE) method. t-SNE aims to find a lower-dimensional projection for each data point such that the conditional probabilities in the projected space are as close as possible to those of the original space (measured by Kullback-Leibler divergence [Kullback and Leibler, 1951]).

Interpreting CNN representations. To understand the representations learned by CNNs, Zeiler and Fergus [2014] discussed how to visualize them via deconvolutional layers. They also presented a method called occlusion maps, such that a sliding window in the image is occluded and the predicted label of the image by the CNN model is checked to see whether that region is diagnostic and important to identify the correct label.

Simonyan et al. [2013] presented a simple guided gradient backpropagation approach for identifying the salient points of the objects with a single forward pass. Compared to occlusion maps, it is computationally more efficient. However, this approach does not point out to the full object extent. Therefore, the output salient points from this approach are used as input to a classical background/foreground segmentation method for object segmentation in natural images. In our case, the segmentation of glyphs from a glyph-block with classical approaches is especially challenging, since neither color nor texture are discriminative for glyphs.

Zhou et al. [2016] introduced class activation maps (CAM) for capturing the full spatial support of objects and not only few salient points. The CAM approach requires to introduce an average pooling layer to model structure. To avoid that, another method called Grad-CAM [Selvaraju et al., 2016] has been proposed as a generalization of CAM. As such, it does not require modifying the CNN model to visualize the activation maps, and it can be applied to any type of CNNs, even to pretrained ones without the need of re-training.

Beyond classification, interpretability is fundamental for domain experts, who need to understand what the method does and match this understanding with their own knowledge. Therefore, we adopted the Grad-CAM approach for illustrating the discriminative parts of the glyphs for the trained model. We discuss the interpretability of the learned representation by the CNN model in a crowdsourcing-based experiment.

6.2 Glyph Visualization Using t-SNE

The analysis process is illustrated in Fig. 6.1. First, for each glyph, a visual representation is computed. As visual representations, we use either a standard knowledge-driven (bag-of-words representation on HOOSC local descriptors, i.e. HOOSC-BoW) or a data-driven representation (activation outputs from a deep CNN). Second, we performed dimensionality reduction on these visual representations of the glyph collection to generate the visualization. The main steps are described in Section 6.2.2 and 6.2.3.

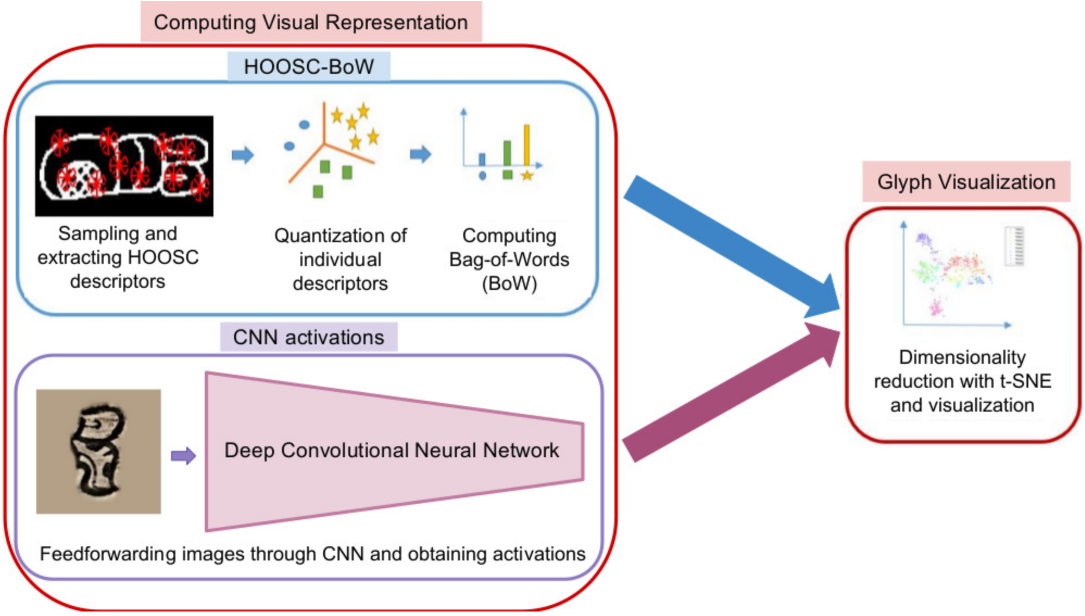

Computing Visual Representation
HOOSC-BoW
Sampling and extracting HOOSC descriptors
Quantization of individual descriptors
Computing Bag-of-Words (BoW)
CNN activations
Deep Convolutional Neural Network
Feedforwarding images through CNN and obtaining activations
Glyph Visualization
Dimensionality reduction with t-SNE and visualization

Figure 6.1 – Overall flow for visualization with t-SNE.

6.2.1 Datasets

We illustrate our visualization pipeline on two individual Maya glyph datasets.

Monument Data: We use a subset of hand-drawings (630 samples from 10 classes, as marked with red in Fig. 2.7) [Roman-Rangel et al., 2011a], corresponding to syllabic glyphs inscribed in monuments. The details of this dataset is presented in Section 2.4.

Thompson Catalog: We use 1487 glyph variants cropped from the [Thompson and Stuart, 1962] catalog. These variants belong to 814 categories and are divided as main sign and prefix/suffix groups in the catalog.

Macri-Vail Catalog: We use 1426 glyph variants across 860 categories from the [Macri and Vail, 2008] catalog. These variants include samples from the [Macri and Looper, 2003] catalog that focuses on hand-drawings of glyph inscriptions from the classic period. These variants from inscriptions have thinner contour and usually more complex than the variants taken from codices.

Codex Blocks: 780 glyph-blocks from codices were cropped and binarized in our joint study with Roman-Rangel et al. [2016]. In this set, there as 12 labels in which the label of a block is defined as combination of its constituent glyph labels.

Individual Codex Glyphs: This set consists of the 9K glyph segmentations from the crowd-sourcing process described in Chapter 4.

6.2.2 Visual Feature Representation

We based our visualizations on both knowledge-driven and data-driven visual representations. The details of these representations are given below.

Knowledge-Driven Representation

The HOOSC has been described earlier in Section 3.3.2. As a reminder, it is computed in two main steps (Fig. 3.2). First, the orientations of a set of sampled points are computed. Secondly, for a given sampled position, the histogram of local orientations are computed using a small number N_a of angle bins forming a circular grid partition centered at each point. The HOOSC descriptor is obtained by concatenating all histograms, and applying per-ring normalization. Basic parameters are the spatial context *sc*, defining the extent of the spatial partition; the number of rings N_r; and the number N_s of slices in a ring. With $N_a = 8$, $N_r = 2$, $N_s = 8$, HOOSC has 128 dimensions. We have used HOOSC for usual retrieval and categorization tasks [Can et al., 2016a; Hu et al., 2015].

Data-Driven Representation

For a data-driven representation, we used deep Convolutional Neural Networks pretrained on ImageNet [Deng et al., 2009]. As the pretrained networks, we chose AlexNet [Krizhevsky et al., 2012]. To extract the visual representations from this pretrained CNN, we feedforwarded our glyph images through the network, and obtained the activation outputs from the 5^{th} convolutional layer (conv-5), and the penultimate fully-connected layer (fc-7).

6.2.3 Dimensionality Reduction: t-SNE

In our application, first, we project the visual representations (either the HOOSC-BoWs or the pretrained CNN activations) to a 30-dimensional space using PCA. Then, we applied t-SNE to these projections to get 2-dimension mapping. t-SNE keeps track of the local structure of the data as it optimizes the clusters globally. The perplexity parameter of the Gaussian kernel employed during the t-SNE optimization is empirically set as 30.

6.2.4 Results and Discussion

Below, we discuss the visualization of several glyph corpora using t-SNE.

Visualizing Glyph Monument Corpus Structure

Fig. 6.2 shows results on the monument corpus. The region encoded in the visual descriptor varies from almost the whole glyph ($sc = 1/1$) to small local parts ($sc = 1/8$). One question is

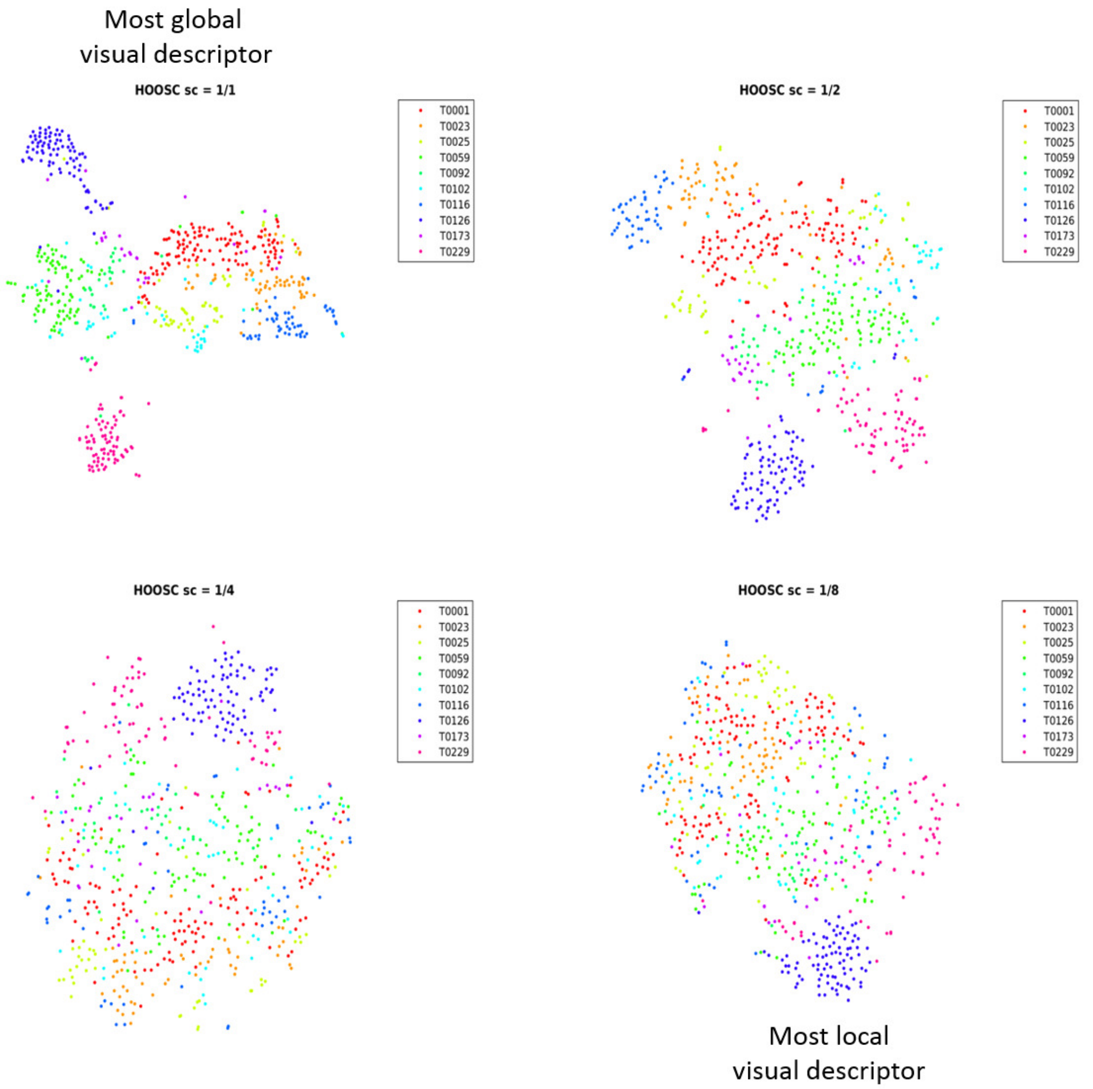


Figure 6.2 – Monument data: t-SNE plots with the visual representations obtained at 4 different spatial context levels.

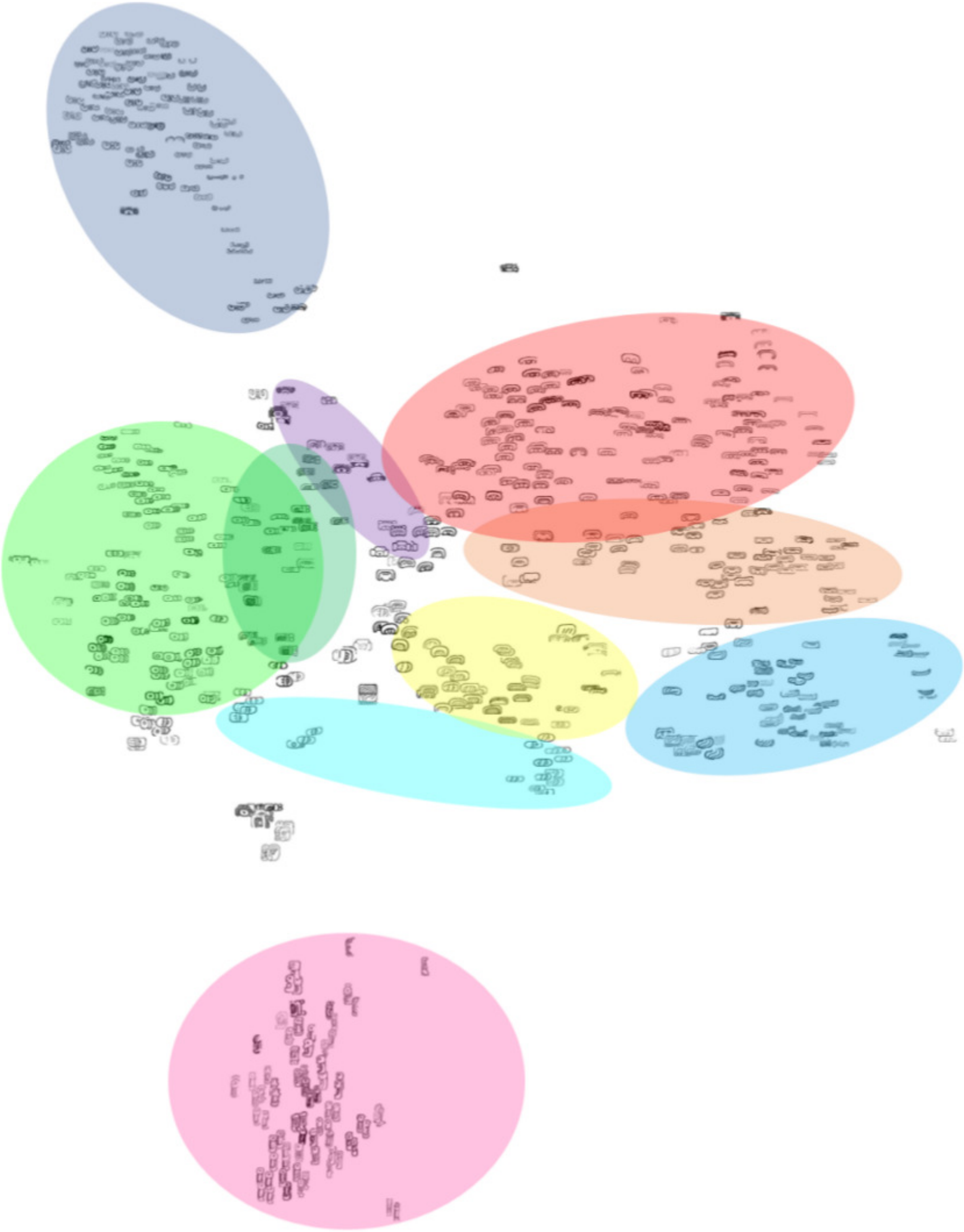


Figure 6.3 – Monument data: Visualization of all class samples with the most global HOOSC descriptor ($sc = 1/1$).

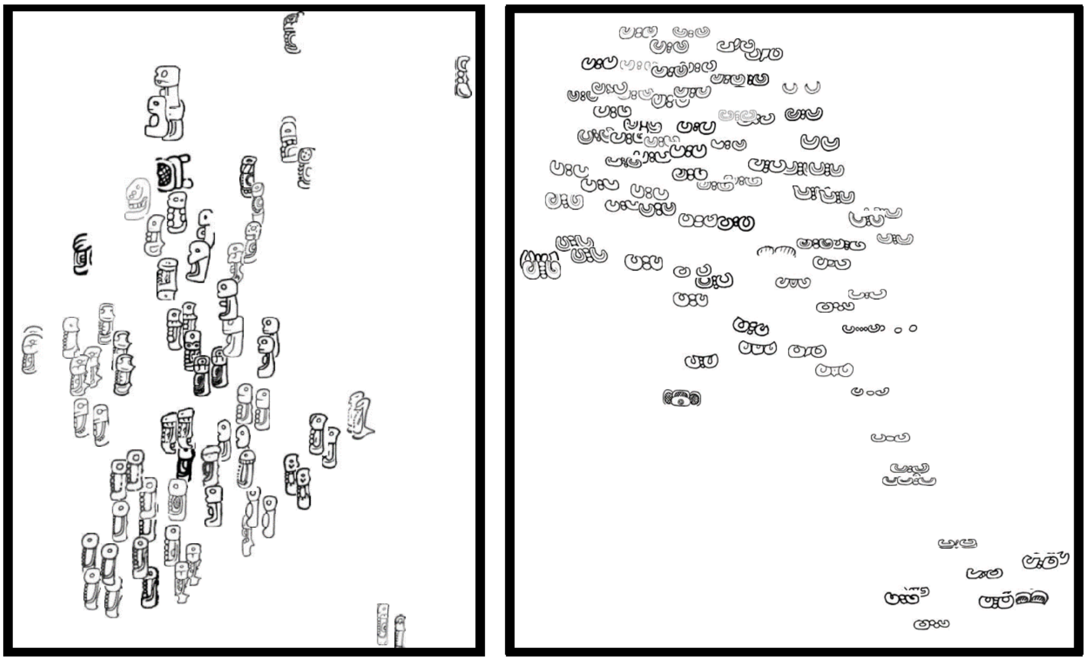


Figure 6.4 – Monument data: Close-up of two clusters (T229 on the left and T126 on the right), corresponding to navy and magenta clusters in Fig. 6.2 with the most global HOOSC descriptor ($sc = 1/1$).

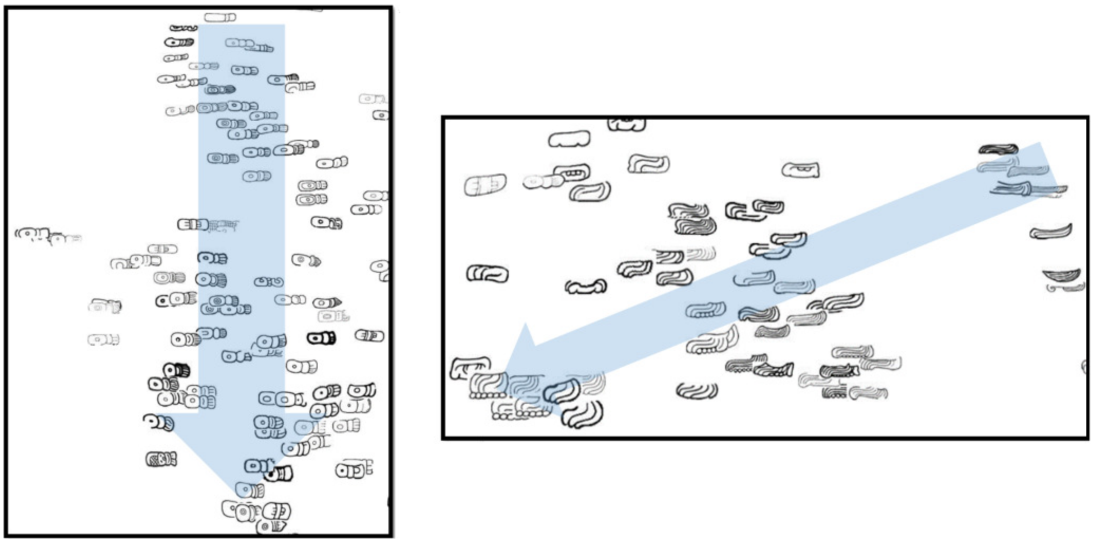


Figure 6.5 – Monument data: Close-up of two clusters (T59 on the left and T116 on the right), which exhibit smooth transitions between samples corresponding to geographic or temporal variations.

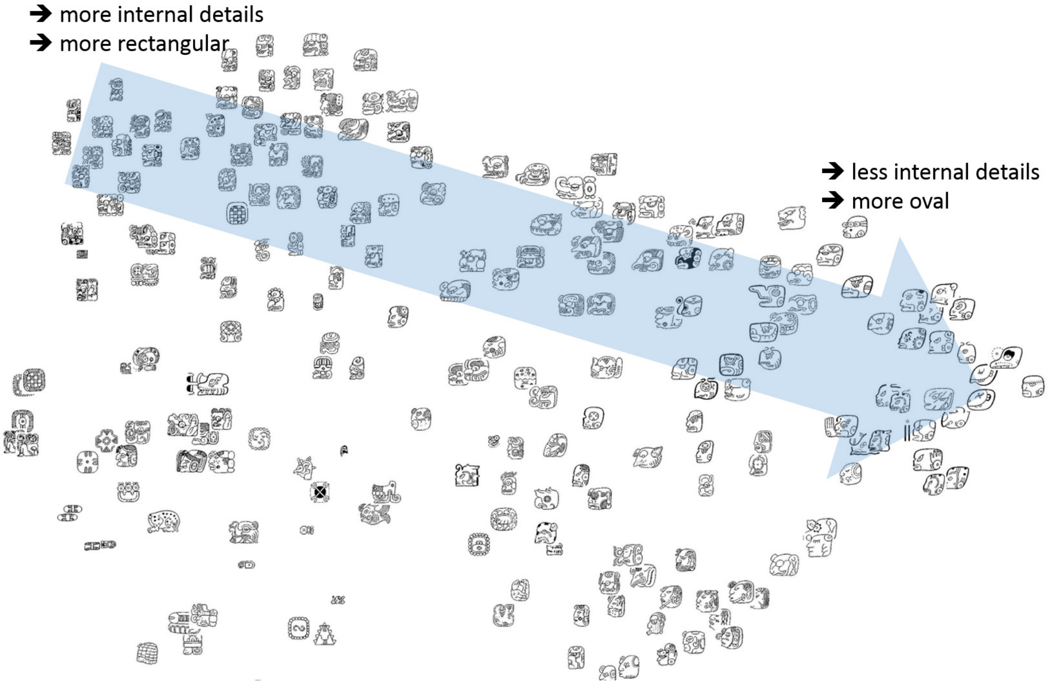
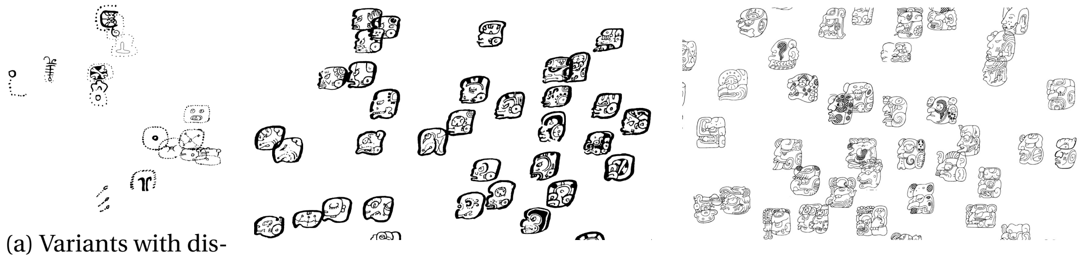


Figure 6.6 – Catalog data: A visual cluster of main signs from the Thompson catalog, with the most global HOOSC descriptor ($sc = 1/1$). Many of them are impersonated main signs that corresponds to gods or animals. In this part of the visualization, the upper left part has more visually complex variants than the rightmost samples.



(a) Variants with discontinuous contour.

(b) Codices head variants.

(c) Monument head variants.

Figure 6.7 – Catalog data: three clusters in the t-SNE visualization of Macri-Vail glyph variants. (a) The variants with discontinuous contour, (b-c) the variants resembling a head (coming from codices and monument inscriptions, respectively).

how spatial context influences the visualization of the representation. Regarding the visual clusters, with the most global representation ($sc = 1/1$), our method extracts more distinct clusters, e.g. T229 and T126 in Fig. 6.4 (navy and magenta clusters in Fig. 6.2 and 6.3). Please refer to Fig. 6.3 for roughly-colored clusters on the actual glyph images. As the descriptor gets more local, the categories with common patterns mix up (Fig. 6.2). Yet, our method is able to capture meaningful common local parts and maps the samples based on these elements, i.e. parallel lines, hatches, and circles.

For Maya epigraphers in our team, a more neatly differentiated grouping of signs, such as that resulting from HOOSC with $sc = 1/1$ is preferable. However, some work to understand the effects of parameter choice is required to obtain groupings that make more epigraphic sense. Clearer "borderlines", less "outliers", and less "intrusive" signs (e.g. T25 and T1) within each cluster would be desirable. Our results in this regard are preliminary, and would require further work with epigraphers.

Another important epigraphic point is that we observe interesting visual transitions between samples of the categories. Fig. 6.5 shows examples from category T59 (left) and T116 (right), which illustrate a smooth dilation of samples in one direction. These kind of observations are interesting for archeologists, since they might correspond to modification of the glyph signs over time or place.

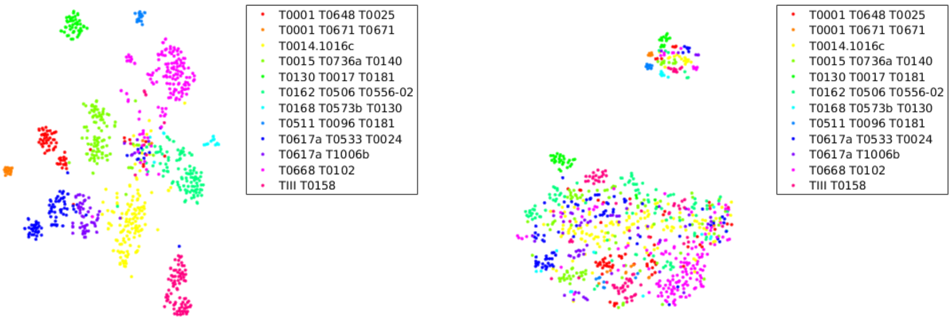
Visualizing Glyph Variants from Catalogs

From the visualization of glyph variants in Thompson's catalog with the largest spatial context level ($sc = 1/1$), we observe that visually similar categories are grouped together, while exhibiting smooth transitions. These transitions may correspond to some characteristics of the data. Fig. 6.6 shows a cluster of personified main signs in which the degree of visual internal detail decreases in the indicated direction. We also observe separate visual clusters for hatched, horizontal and vertical glyphs.

Furthermore, we visualized Macri-Vail catalog sign variants according to the pretrained CNN representations (specifically, activation output from the 5^{th} pooling layer of AlexNet). In Macri-Vail catalog, the drawings of monumental signs are thinner than the signs taken from codices. We observe this main grouping in our visualization as well. Fig. 6.7 shows three example clusters in the visualization of the Macri-Vail catalog variants. Similar to the Thompson variants case, the signs with similar outline or characteristics, e.g. head signs, elongated signs, signs with three separate components, and discontinuous contours with dots, are grouped together.

Visualizing Codical Glyph-Blocks

In our joint study with Roman-Rangel et al. [2016], we studied retrieval and indexing of a set of binarized glyph-blocks from the three Maya codices (Dresden, Madrid, and Paris codices).



(a)

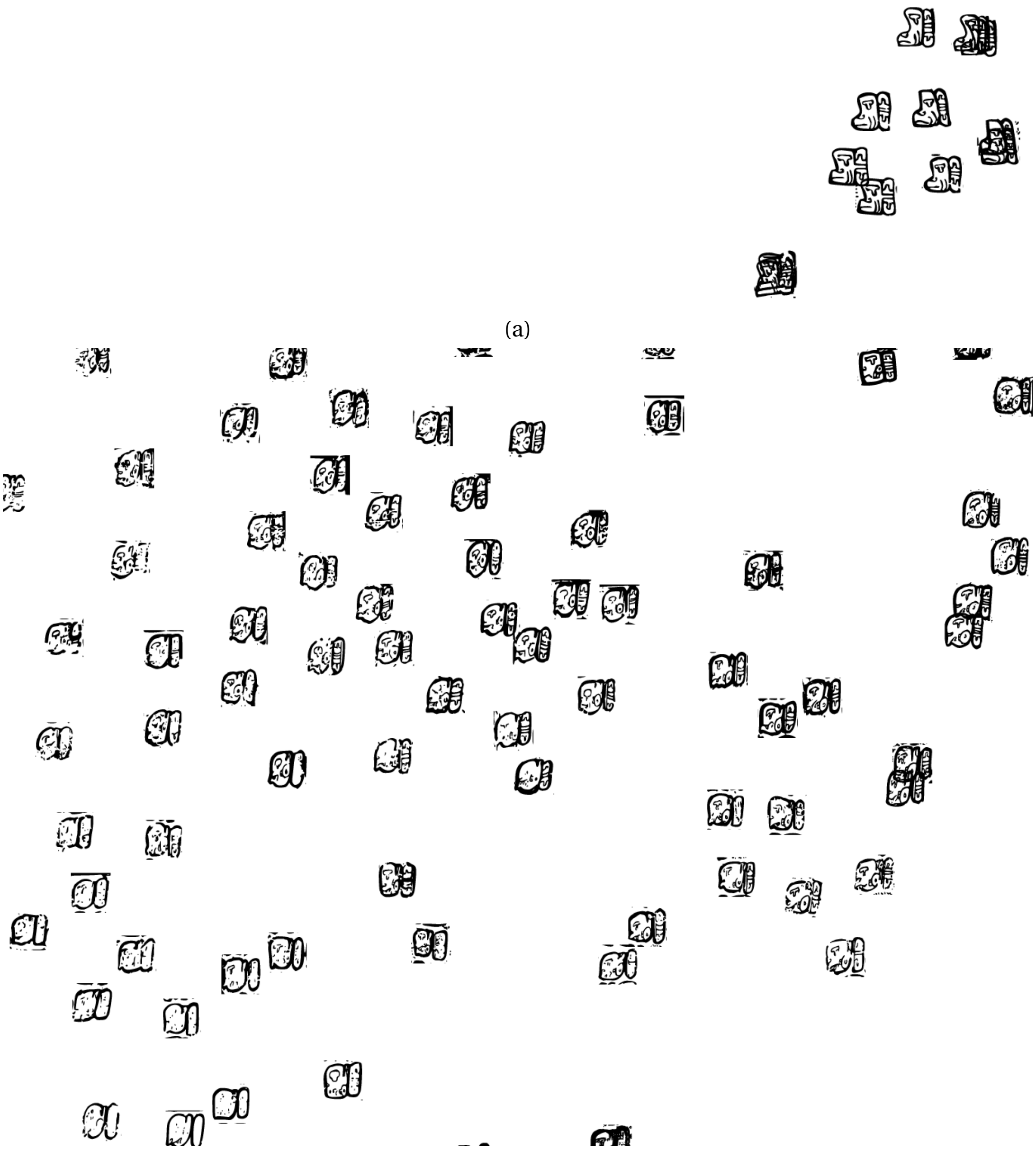
(b)

Figure 6.8 – Codical glyph-blocks: t-SNE plots based on (a) the 5^{th} convolutional layer (conv5) activations, and (b) the penultimate fully-connected layer (fc-7) activations of the pretrained AlexNet.

Employing t-SNE method improved the retrieval performance of both the HOOSC-BoW and the CNN representations (specifically, the activation output from the 5^{th} convolutional layer -conv5- or the penultimate fully-connected layer -fc7- of VGG-16). Futhermore, the conv5-tSNE representation outperformed other representations. In this retrieval task, we also evaluated the conv5 activation outputs from AlexNet together with the t-SNE method, and observed that they performed similar to the conv5-tSNE representations from the VGG-16. Below, to be consistent with the other related figures in this chapter, we showed the block visualizations obtained with the conv5 representations from the pretrained AlexNet.

Fig. 6.8 presents the t-SNE scatter plots of the codical blocks according to (a) the 5^{th} convolutional layer (conv5) activations, and (b) the penultimate fully-connected layer (fc-7) activations of the pretrained AlexNet. These scatter plots show that the conv5-tSNE representation is able to separate blocks better than the fc7-tSNE representation. We observed that the small cluster on the top in the scatter plot of fc7-tSNE representation contains the manually-cleaned training set of blocks. On the other hand, the larger bottom cluster contains automatically-binarized block images that contain “salt-and-pepper” type of noise [Roman-Rangel et al., 2016]. Therefore, the conv5-tSNE representation proves itself to be more robust to the noise in the data compared to the fc7-tSNE representation. Furthermore, the mixed-labeled cluster in the middle of all the clusters in Fig. 6.8a corresponds to the damaged partial glyphs. Among these damaged blocks, the blocks composed of the same glyphs are close to each other.

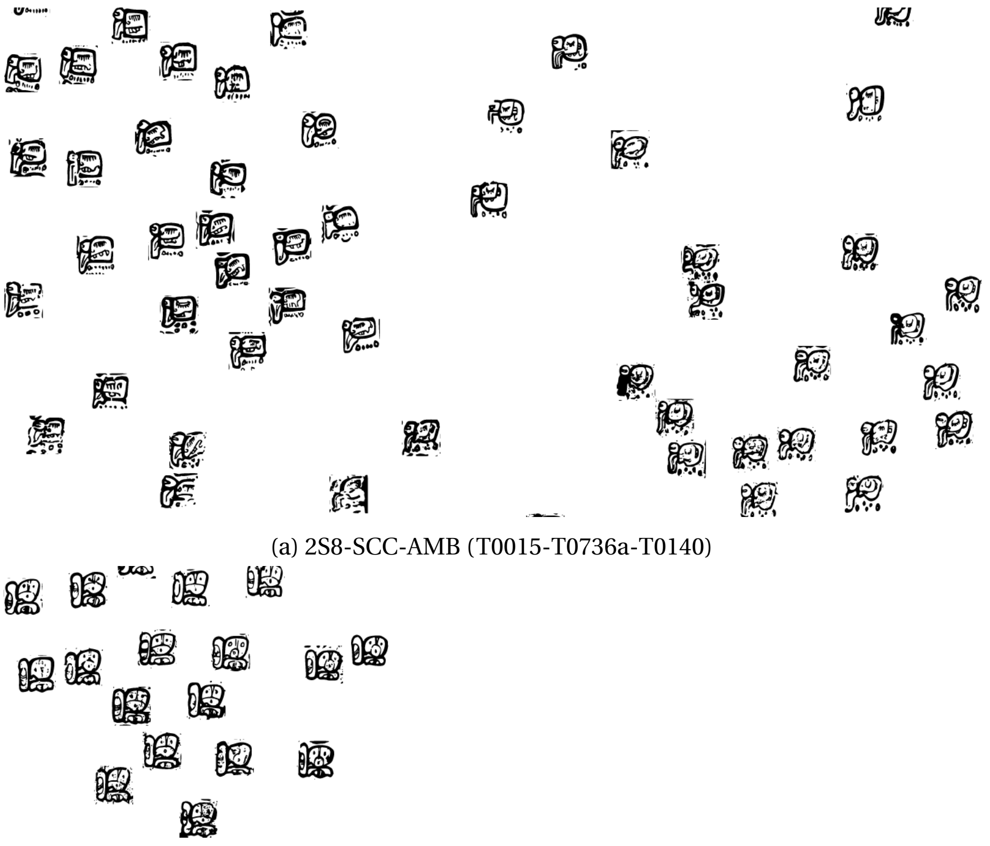
Fig. 6.9 shows two clusters of the MZ9-1B2 blocks (or T0668-T0102 with Thompcon codes, representing “Rain God”). Fig. 6.9b shows the main cluster of these blocks, whereas Fig. 6.9a shows an outlier cluster with deformed contour of the main sign (MZ9 or T0668). Despite the high deformation level in the outliers, these block instances are positioned close (just above the main cluster). This point highlights the accuracy of the shape representation and the dimensionality reduction method. Another interesting observation in Fig. 6.9b is that the



(a)

(b)

Figure 6.9 – Codical glyph-blocks: (a) A small cluster of the MZ9-1B2 blocks (or T0668-T0102 with Thompson codes, representing "Rain God") that have deformed contour. (b) The main cluster of the MZ9-1B2 glyph-blocks. Damage level of the blocks decreases from top-right to bottom-left.



(a) 2S8-SCC-AMB (T0015-T0736a-T0140)

(b) 1M2-AMB-1M4 (T0617a-T0533-T0024)

Figure 6.10 – Codical glyph-blocks: Two clusters of glyph-blocks that have sub-clusters according to circular (Dresden Codex) vs. rectangular (Madrid Codex) styles. The cluster of glyph-blocks composed of (a) 2S8-SCC-AMB glyphs (or T0015-T0736a-T0140 with Thompson codes), and (b) 1M2-AMB-1M4 glyphs (T0617a-T0533-T0024). In (a) the main signs (SCC) are more rectangular in the top-left sub-cluster compared to the bottom-right cluster. Similarly, in (b), the signs are more rectangular in the bottom-right sub-cluster compared to the top-left cluster.

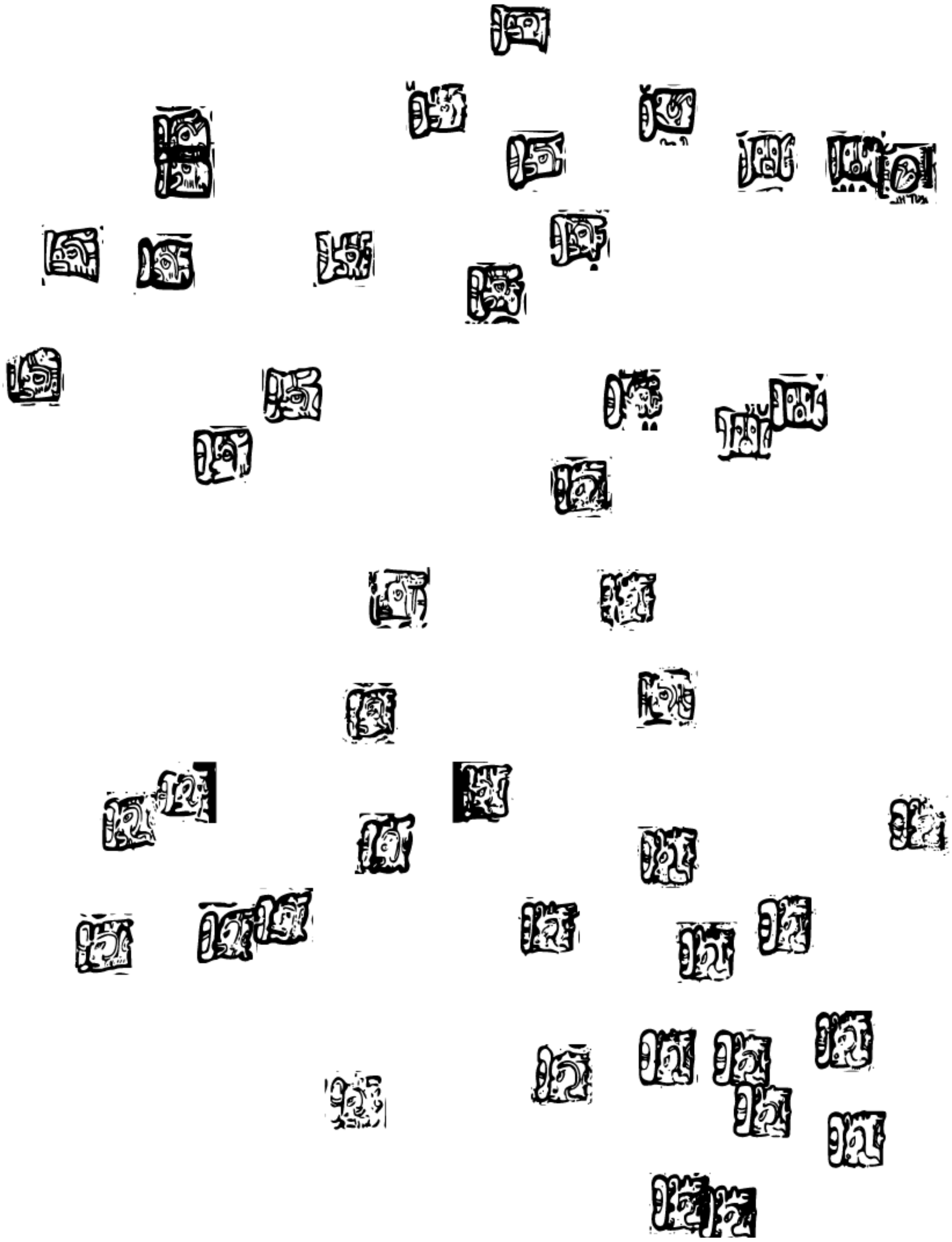


Figure 6.11 – Codical glyph-blocks: Stylistic differences on the main sign (PE8 or T1006, “Maize God”). Examples at the bottom show more details for the top part of the glyph (“hairdress”).

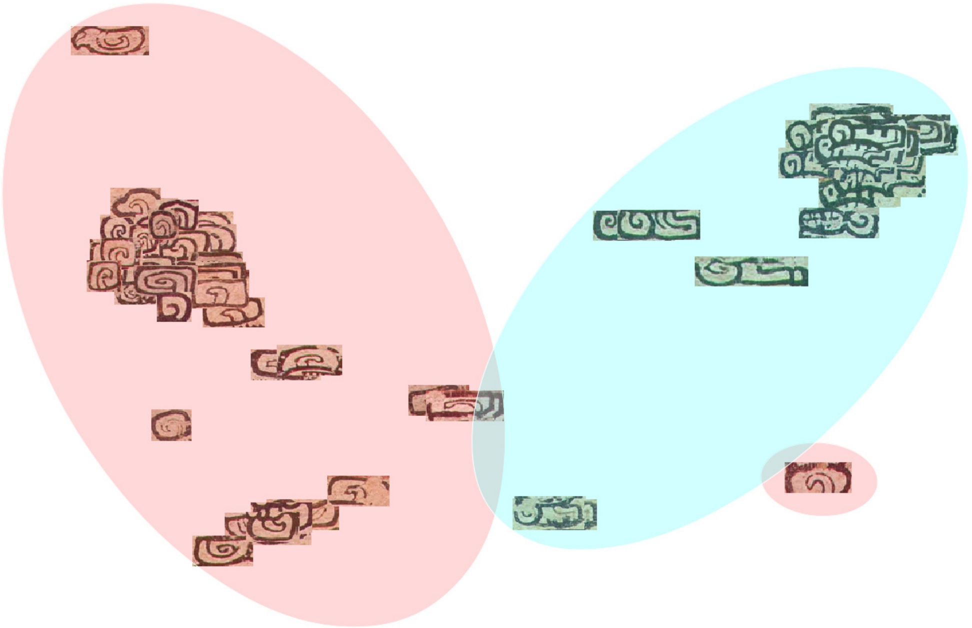


Figure 6.12 – Crowdsourced codical individual glyphs: Partial visualization of the 2S2 glyphs via t-SNE algoritm shows the separation of glyphs corresponding to two different variants (see Fig.4.18b, blue cluster for the first, pink cluster for the second variant).

damage levels of the blocks decrease from top-right to bottom-left.

Fig. 6.10 and 6.11 illustrate the stylistic differences among the instances of the three blocks. In Fig. 6.10a, we observe two main sub-clusters of the glyph-blocks with 2S8-SCC-AMB glyphs (or T0015-T0736a-T0140 with Thompson codes). At the bottom-right sub-cluster of Fig. 6.10a, the main signs (SCC) exhibit a circular style with a dent at the bottom. This style is mostly observed in Dresden Codex. On the other hand, at the top-left sub-cluster of Fig. 6.10a, the main signs are more rectangular (Madrid Codex style). This trend is observed in the other clusters as well. In the case of the 1M2-AMB-1M4 blocks (T0617a-T0533-T0024) as shown in Fig. 6.10b, the signs in the bottom-right sub-cluster is more rectangular than the blocks in the top-left sub-cluster. Furthermore, we observe two different variants of the bottom glyph (AMB) in the sub-clusters of Fig. 6.10a. The AMB glyphs in the top-left of Fig. 6.10a are composed of a linear series of dots, whereas the AMB glyphs in the bottom-right have a triangular composition of dots in the middle.

Visualizing Individual Codical Glyphs

After obtaining the individual glyph segments with the crowdsourcing approach described in Chapter 4, we analyzed the visual clusters in each glyph category by the help of the t-SNE method. Here, we present two cases, namely visual clusters of 2S2 in Fig. 6.12 and SCC in Fig. 6.13. Fig. 6.12 shows the separation of 2S2 glyphs corresponding to two different variants

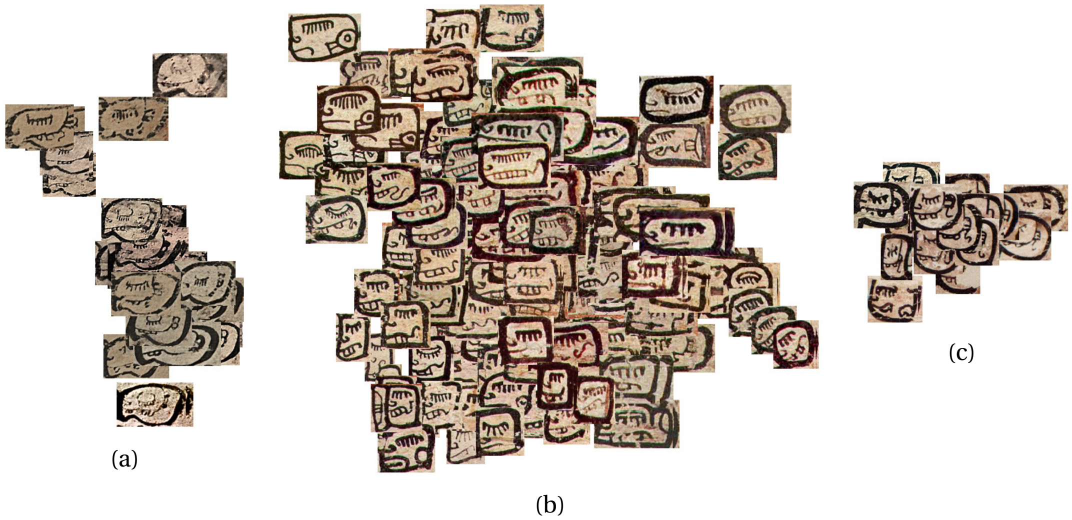


Figure 6.13 – Crowdsourced codical individual glyphs: Visual clusters of glyphs belonging to SCC ("Death") category from (a) Dresden Codex, (b) Madrid Codex, and (c) Paris Codex.

(see Fig.4.18b for these variants). Blue cluster corresponds to the samples that belongs to the first variant, and pink cluster indicates the glyphs belonging to the second variant. Fig. 6.13 illustrates that the instances of SCC category from the different codices are clustered separately as they exhibit stylistic differences. We observe that the instances from Dresden codex have a circular general outline with a dent at the bottom left. Similar to the Dresden instances, the samples from Paris Codex are circular, but smaller, whereas the instances from Madrid Codex are more rectangular.

6.2.5 Conclusion

Our goal in this study was to help Digital Humanities scholars to visualize data collections not as isolated elements, but in context (visually and semantically).

Even though early catalogs are built based on visual similarities, i.e., Thompson [Thompson and Stuart, 1962] or Zimmerman [Zimmerman, 1956] relied on graphic cards to study similar patterns and spatial distributions, the categorization methods were poorly understood and were not easy to reconfigure. Furthermore, due to the limited knowledge at the time about semantics and sign variants, these catalogs turned out to be inaccurate or outdated. Similarly, Gardiner's list [Gardiner, 1957] is insufficient to elucidate sign variability in the "Book of The Dead" [Budge, 1901].

With the proposed tool, however, considering details at different scales as semantic/diagnostic regions in the visualization could help archaeologists to discover semantic relations. In this way, overlapping notions such as "colors", "cardinal directions" and specific toponyms from earthly, heavenly, or underworld realms could be studied in greater detail.

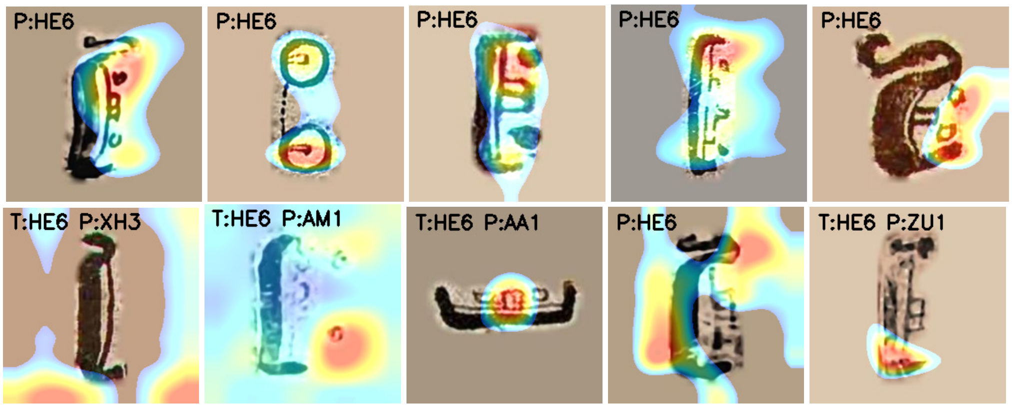


Figure 6.14 – Success (top row) and failure examples (bottom row) of activations of discriminative parts for sign HE6 with the 50-class SaN-S model. Positive activation from the model for each input image is illustrated as a colored intensity map (referred as heatmap) over the input image. In this map, blue indicates a weak, yellow indicates a medium-level, and red indicates a strong positive activation. The heatmaps are visualized via Grad-CAM based on the predicted class ("P") by the model for the given input image. If the model made a false prediction, the true class of the glyph image is also indicated (as "T"). The class labels are given in Macri-Vail catalog codes.

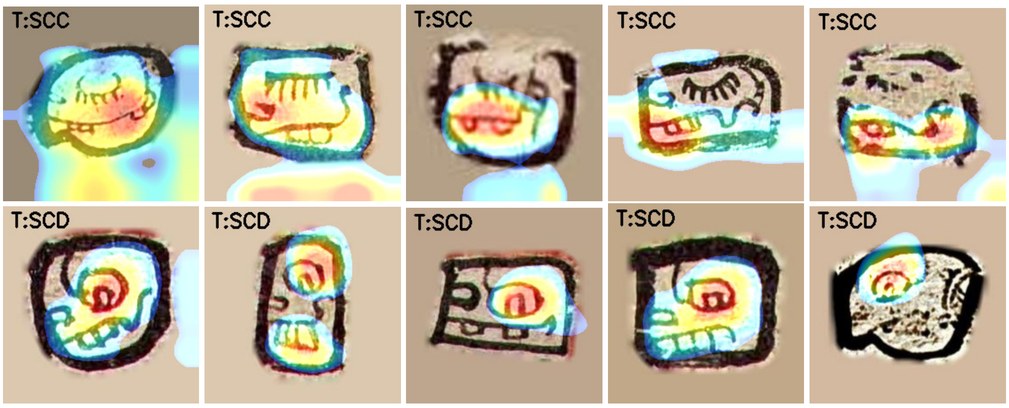


Figure 6.15 – The SCC (top row) and SCD examples (bottom row) and the activations of discriminative parts for the 50-class SaN-S model. The heatmaps are visualized via Grad-CAM method based on the true class ("T").

Finally, illustrating all variations with different visual focus in a fast and quantitative manner brings out the characteristics of signs. This could help experts match samples from various sources (i.e. monuments, codices, and ceramic surfaces) to corpus data more efficiently; and trigger the decipherment of less frequent and damaged signs. Hence, our work could potentially contribute towards producing a more accurate and state-of-the-art sign catalog.

6.3 Visualization of Diagnostic Parts and Interpretability

6.3.1 Grad-CAM as Visualization of Diagnostic Parts

To understand the learned CNN representations, we utilized both gradient backpropagation [Simonyan et al., 2013] and Grad-CAM [Selvaraju et al., 2016] methods. With these methods, we visualized where the salient and the discriminative parts of the glyphs are for the models studied in Chapter 5. Another use-case of such a class-activation visualization is to localize the glyphs in glyph-blocks (cluttered scenes) given the model trained on the desired glyph class.

Salient point visualization. After a single forward pass of the input through the network, partial derivatives of predicted class score w.r.t. pixel intensities are backpropagated and visualized Simonyan et al. [2013]. This corresponds to visualizing the importance of input pixels such that the predicted class score is affected the most in case of a change in the input intensities.

CAM. Class Activation Mapping Zhou et al. [2016] is defined on a CNN that ends with a block of "convolutional layer → global average pooling layer → softmax layer". Hence, this visualization approach requires re-training of the weights after changing the CNN architecture, i.e. by replacing the layers after the last convolutional layer with Global Average Pooling (GAP) and a softmax layer. Then, the class score is obtained by a forward pass of the activations from the last convolutional layer:

$$y^c = \sum_k w^c \frac{1}{Z} \sum_i \sum_j A^k_{ij}, \tag{6.1}$$

The localization map $L^c \in \mathbb{R}^{u \times v}$ for class c with input width u and height v is computed as a linear combination of k feature map activations A^k and the re-trained weights w^c_k between the Global Average Pooling layer and the softmax layer:

$$L^c = \sum_k w^c_k A^k. \tag{6.2}$$

Grad-CAM. Being a generalization of CAM (except the final ReLU operation), GradCAM does not require a change in the CNN architecture and it is applicable to other neural network architectures as well Selvaraju et al. [2016].

Fig. 6.16 illustrates the GradCAM method. Essentially, first, the backpropagated gradients $\frac{\partial y^c}{\partial A^k_{ij}}$ are global-average-pooled and the importance weights α^c_k of feature map activations A^k are obtained:

$$\alpha_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}, \tag{6.3}$$

These weights correspond to the re-trained weights w_k^c in CAM approach. Secondly, the linear combination of these weights and the feature map activations A^k are computed. Finally, this linear combination is passed through a ReLU activation so that only positive activation-gradient combinations are considered.

$$L^c = ReLU\left(\sum_k \alpha_k^c A^k\right). \tag{6.4}$$

In our case, to pay attention only to the most characteristic glyph parts, as a final operation, we eliminated the weak activation-gradient combinations (lower than 0.5) in the localization map.

Table 6.1 illustrates samples from the Thompson (T) and Macri-Vail (MV) catalogs, as well as available expert comments about the diagnostic features of five glyph categories. We demonstrate the salient points obtained via [Simonyan et al., 2013], and the Grad-CAM heatmaps that correspond to the discriminative part of the glyph sample according to the trained model (i.e. the SaN trained from scratch with 50-classes); red means high and blue means low response. We also present the class accuracies of these specific classes obtained with this model. Besides, we present Grad-CAM responses of these glyph categories in the glyph-blocks for both positive

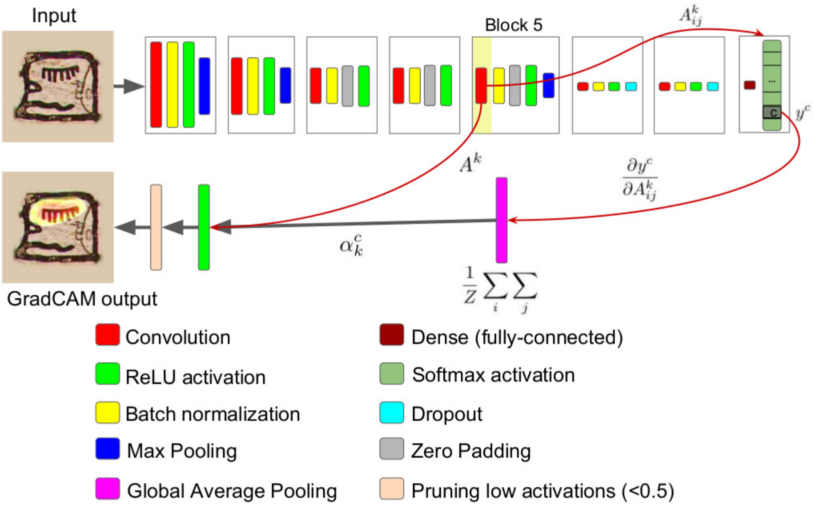


Figure 6.16 – Illustration of the GradCAM method.

(+) cases, and negative, uninformative, or partial-correspondence (-) cases.

Table 6.1 shows that Grad-CAM is often successful at capturing discriminative features for the shown glyphs, and it is also able to localize them in glyph-blocks (column 6). However, it also fails in several cases as the negative examples show (column 7). As Selvaraju et al. [2016] reported, gradient backpropagation is less informative than Grad-CAM method.

Fig. 6.14 depicts examples of discriminative part heatmaps obtained via Grad-CAM according to the predicted class for the HE6 class (first glyph in Table 6.1). The top row illustrates successful visualizations that match the expert comments, whereas the bottom row shows failures of Grad-CAM. Note that all the top row examples are classified correctly, even the second example which is a different variant of the glyph and is not dominant in the training set. On the other hand, in the bottom row, we observe that the absence or different disposition of the two diagnostic dots end up in misclassification or not-so-well-localized heatmaps.

In the Macri and Vail [2008] Maya sign catalog, the SCC sign that means "death or dead" is described as "head with closed eye", whereas the SCD sign that corresponds to "death god" is described as "skull" or "skull with eyeballs". Fig. 6.15 depicts heatmaps of the true class activations of these glyphs (top row for SCC, bottom row for SCD). Notably, for the SCD examples, we observe that the attention of the CNN representation is on the "eyeball", yet for the SCC examples the nose and the teeth around the mouth are also highlighted. This is most likely due to the existence of other categories that showcase the "closed eye with eyelashes" as their main specificity.

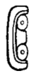
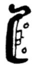
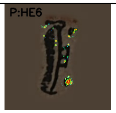
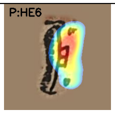
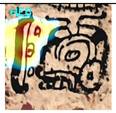
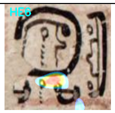
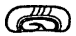
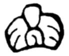
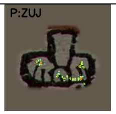
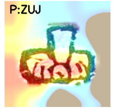
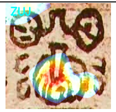
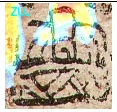
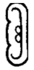
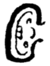
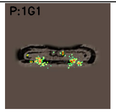
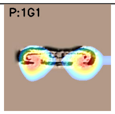
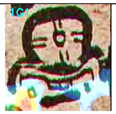
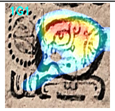
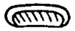
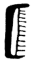
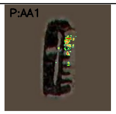
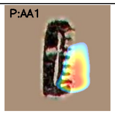
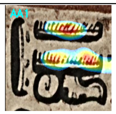
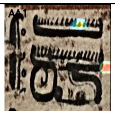
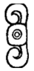
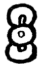
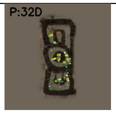
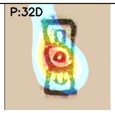
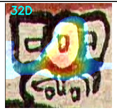
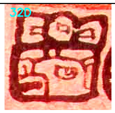
Overall, we see that the use of Grad-CAM is promising to visualize diagnostic features of glyphs. How to systematize the evaluation of the visual explanations produced by deep networks is investigated in a preliminary study described in the next section.

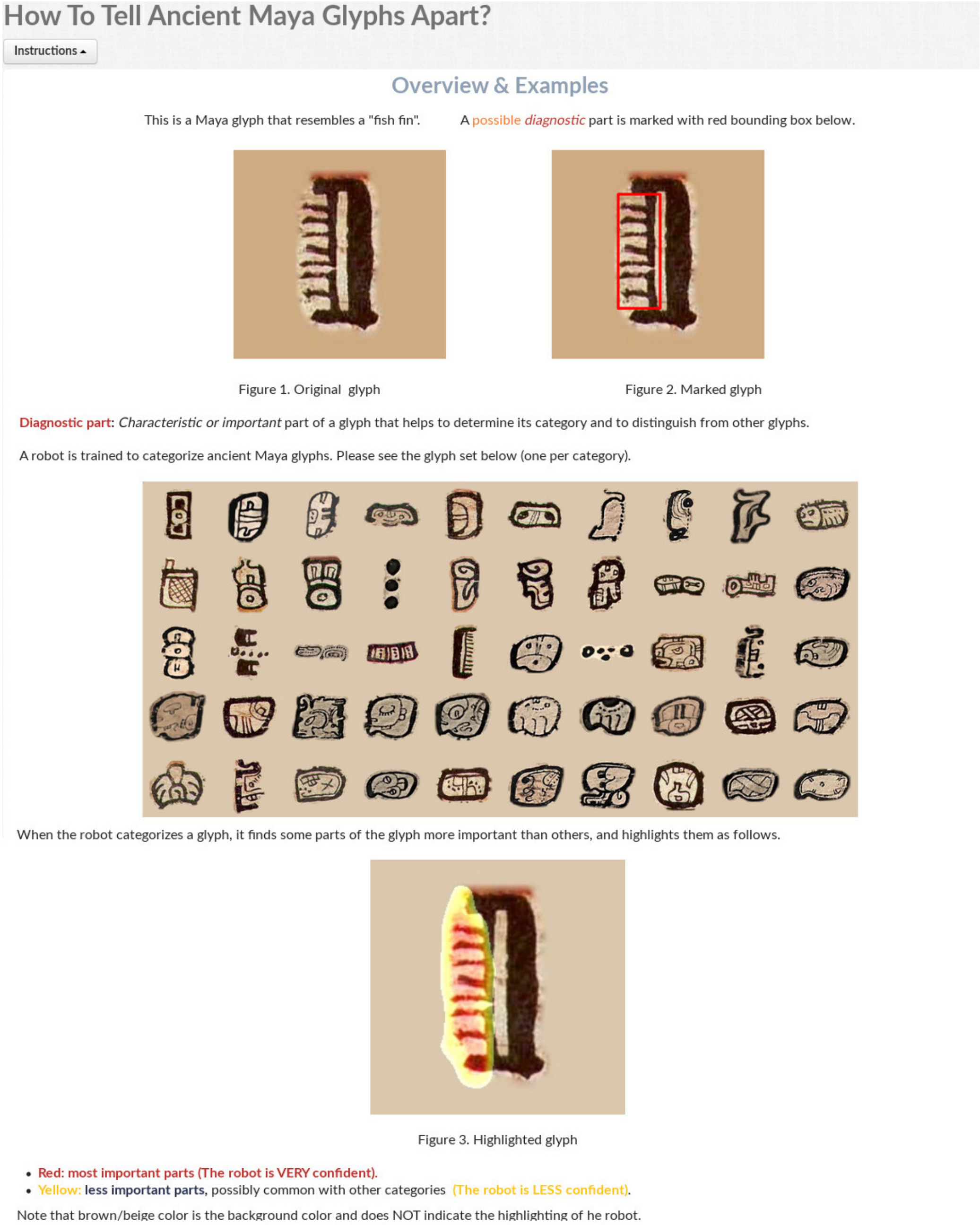
6.3.2 Qualitative Crowdsourcing Analysis on CNN Visualizations

To assess the interpretability of the CNN representations *qualitatively*, we performed a preliminary crowdsourcing study. Specifically, this study is a perceptual comparative analysis of the visual representations of a sketch-specific network (Sketch-a-Net) and a residual network (ResNet-50). These representations were trained on 50 classes of individual glyphs that were obtained via the crowdsourcing process described in Chapter 4. The details of the data treatment and CNN training can be found in Chapter 5.

As opposed to crowdsourcing studies with every-day objects in natural images, we have the challenge of non-experts not having a predefined concept of glyph categories. Thus, in our task design, we prepared detailed instructions and provided supervision to non-experts. Our focus is on whether non-experts' perception is at all aligned with the automatic discriminative models. We also address the research question of which CNN produces more appealing visual explanations according to the crowd (a sketch-driven network or a residual network).

Table 6.1 – Diagnostic features of five Maya signs, commented by epigrapher experts. We present the discriminative parts inferred by the 50-class SaN-S model. The heatmaps are visualized via the guided backpropagation approach and the Grad-CAM method (red is for high, blue is for low response). Column 6 and 7 present the examples of weak localization of corresponding glyphs in the glyph-blocks. (+) indicates a correct correspondence (examples in column 6), and (-) indicates a wrong, uninformative, or partial-correspondence (column 7).

T class	MV class	Diagnostic Feature	Salient Points	Grad-CAM	Grad-CAM in a block (+)	Grad-CAM in a block (-)	Class Acc. (%)
 001	 HE6	An outer C-shaped frame, two circles, the "teeth" attached to inner line of the frame					96.7
 017	 ZUJ	A hook inside the central circle, two to four vertical parallel lines coming up from the central circle.					96.0
 023	 1G1	Two notches/discontinuity at the bottom of its outer shape, inner thin line whose two ends face downwards, the "teeth" elements attached to the inner line.					85.0
 025	 AA1	Parallel elongated lines, framed by a thicker general outline.					96.3
 061	 32D	General outline resembling a necklace that is composed of a precious stone in the middle, and twines on both sides of it.					88.3



How To Tell Ancient Maya Glyphs Apart?

Overview & Examples

This is a Maya glyph that resembles a "fish fin".

A possible *diagnostic* part is marked with red bounding box below.

Figure 1. Original glyph

Figure 2. Marked glyph

Diagnostic part: *Characteristic or important* part of a glyph that helps to determine its category and to distinguish from other glyphs.

A robot is trained to categorize ancient Maya glyphs. Please see the glyph set below (one per category).

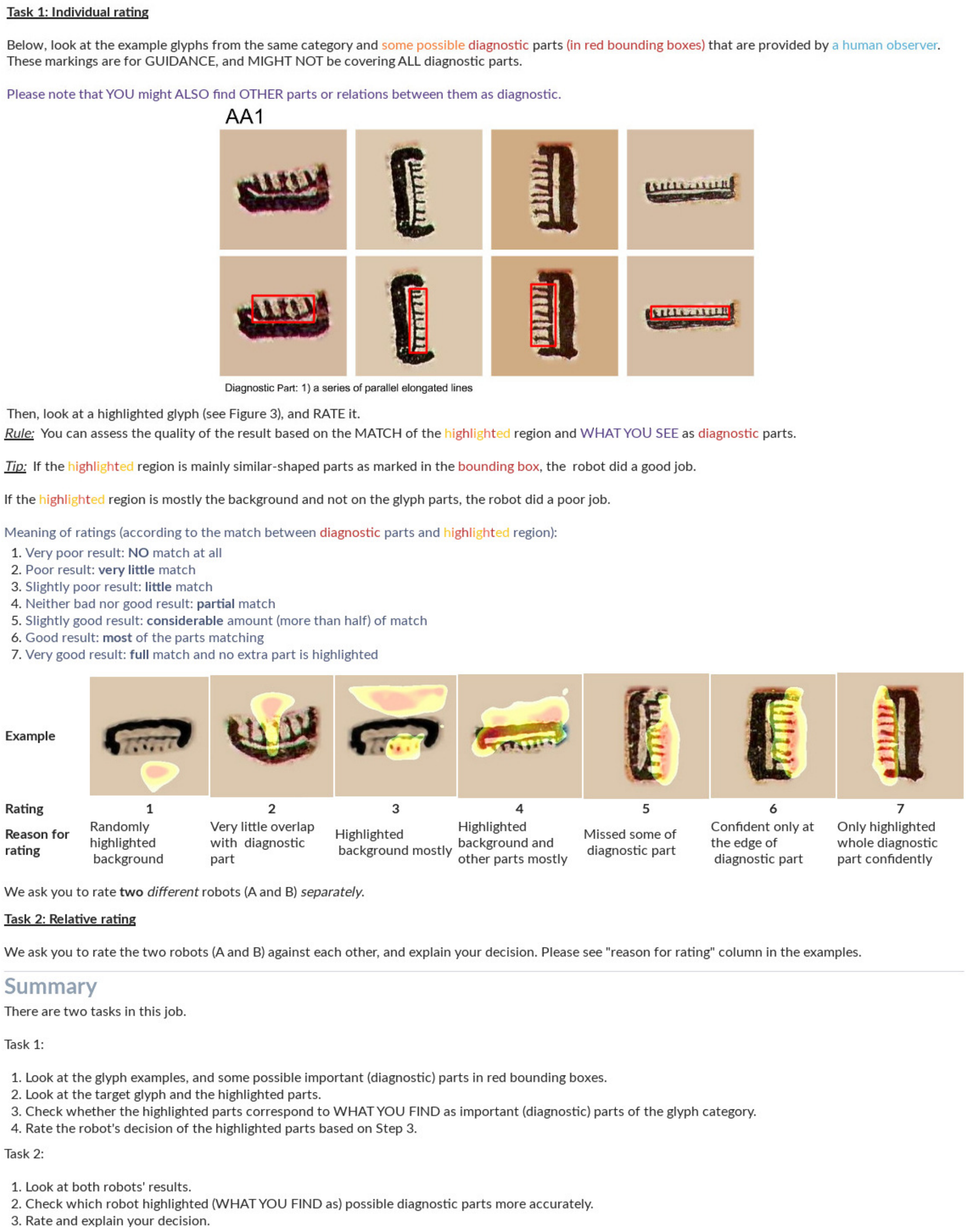
When the robot categorizes a glyph, it finds some parts of the glyph more important than others, and highlights them as follows.

Figure 3. Highlighted glyph

- Red: most important parts (The robot is VERY confident).
- Yellow: **less important parts**, possibly common with other categories (The robot is LESS confident).

Note that brown/beige color is the background color and does NOT indicate the highlighting of he robot.

Figure 6.17 – Introductory part in the instructions of the crowdsourcing task. This part shows a glyph example with its diagnostic part, demonstrates examples from other classes that the CNN had been trained on, and the CNN visualization on the first example glyph.



Task 1: Individual rating

Below, look at the example glyphs from the same category and some possible diagnostic parts (in red bounding boxes) that are provided by a human observer.
These markings are for GUIDANCE, and MIGHT NOT be covering ALL diagnostic parts.

Please note that YOU might ALSO find OTHER parts or relations between them as diagnostic.

Diagnostic Part: 1) a series of parallel elongated lines

Then, look at a highlighted glyph (see Figure 3), and RATE it.
Rule: You can assess the quality of the result based on the MATCH of the highlighted region and WHAT YOU SEE as diagnostic parts.

Tip: If the highlighted region is mainly similar-shaped parts as marked in the bounding box, the robot did a good job.

If the highlighted region is mostly the background and not on the glyph parts, the robot did a poor job.

Meaning of ratings (according to the match between diagnostic parts and highlighted region):

1. Very poor result: **NO** match at all
2. Poor result: **very little** match
3. Slightly poor result: **little** match
4. Neither bad nor good result: **partial** match
5. Slightly good result: **considerable** amount (more than half) of match
6. Good result: **most** of the parts matching
7. Very good result: **full** match and no extra part is highlighted

Example							
Rating	1	2	3	4	5	6	7
Reason for rating	Randomly highlighted background	Very little overlap with diagnostic part	Highlighted background mostly	Highlighted background and other parts mostly	Missed some of diagnostic part	Confident only at the edge of diagnostic part	Only highlighted whole diagnostic part confidently

We ask you to rate **two** *different* robots (A and B) *separately.*

Task 2: Relative rating

We ask you to rate the two robots (A and B) against each other, and explain your decision. Please see "reason for rating" column in the examples.

Summary

There are two tasks in this job.

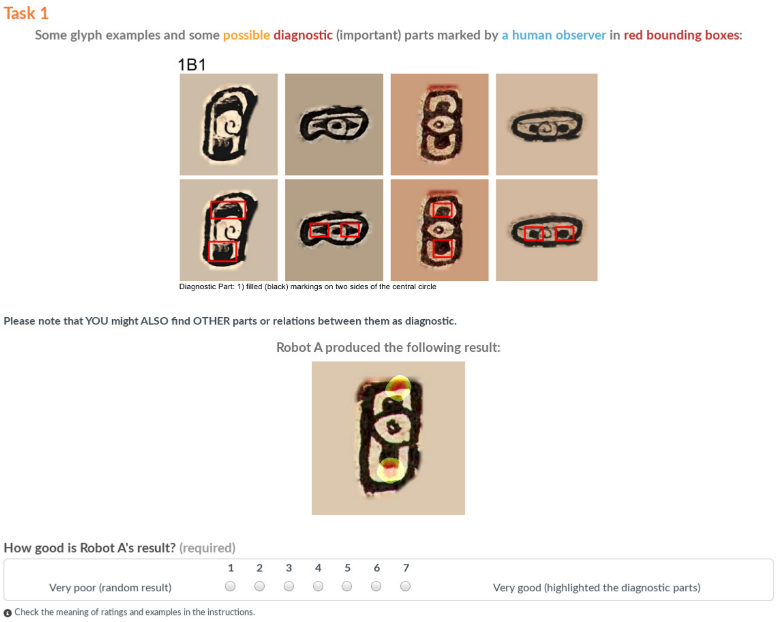
Task 1:

1. Look at the glyph examples, and some possible important (diagnostic) parts in red bounding boxes.
2. Look at the target glyph and the highlighted parts.
3. Check whether the highlighted parts correspond to WHAT YOU FIND as important (diagnostic) parts of the glyph category.
4. Rate the robot's decision of the highlighted parts based on Step 3.

Task 2:

1. Look at both robots' results.
2. Check which robot highlighted (WHAT YOU FIND as) possible diagnostic parts more accurately.
3. Rate and explain your decision.

Figure 6.18 – Instructions on the specific parts of the crowdsourcing task. This part describes the parts of the task, indicates the meanings of the ratings both visually and textually. Finally, it summarizes the steps of the task.



Task 1

Some glyph examples and some possible diagnostic (important) parts marked by a human observer in red bounding boxes:

Diagnostic Part: 1) filled (black) markings on two sides of the central circle

Please note that YOU might ALSO find OTHER parts or relations between them as diagnostic.

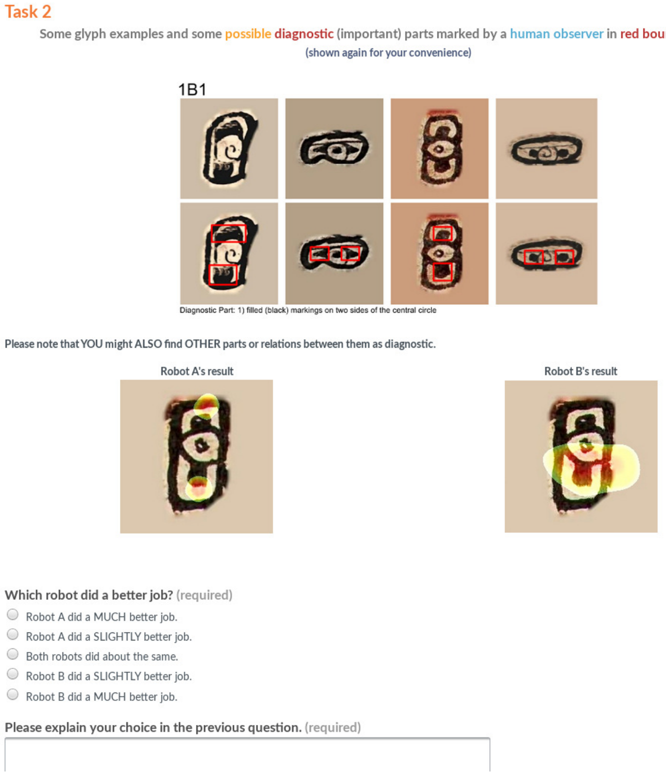
Robot A produced the following result:

How good is Robot A's result? (required)

	1	2	3	4	5	6	7	
Very poor (random result)	○	○	○	○	○	○	○	Very good (highlighted the diagnostic parts)

Check the meaning of ratings and examples in the instructions.

(a) First task: Individual rating.



Task 2

Some glyph examples and some possible diagnostic (important) parts marked by a human observer in red bounding boxes:
(shown again for your convenience)

Diagnostic Part: 1) filled (black) markings on two sides of the central circle

Please note that YOU might ALSO find OTHER parts or relations between them as diagnostic.

Robot A's result

Robot B's result

Which robot did a better job? (required)

- ○ Robot A did a MUCH better job.
- ○ Robot A did a SLIGHTLY better job.
- ○ Both robots did about the same.
- ○ Robot B did a SLIGHTLY better job.
- ○ Robot B did a MUCH better job.

Please explain your choice in the previous question. (required)

(b) Second task: Relative rating.

Figure 6.19 – Two parts of the crowdsourcing task: (a) individual rating of a CNN visualization, (b) relative rating of two CNN visualizations against each other.

Data

In this preliminary study, we limited our dataset to 10 glyphs from 10 classes. We computed the Grad-CAM results of two networks for these randomly-chosen 100 glyphs. To make the visualizations more selective, we eliminated the weak activations in these mappings (empirical threshold is set as 0.5 in the range of $[0,1]$).

In an attempt to understand the visualizations on this set of chosen glyphs, as a preliminary check, we performed a pixelwise comparison between these visualizations and the manually-marked groundtruth masks. We observed that the pruned ResNet-50 visualizations overlap with the groundtruth masks in all cases, whereas the Sketch-a-Net visualizations overlap with the groundtruth masks for 98 out of the 100 glyphs. Furthermore, the ResNet-50 visualizations were more diffused with higher average recall (0.754 vs. 0.582) and lower average precision values (0.255 vs. 0.431).

Over this set of chosen glyphs, to make the perceptual task easier for non-expert observers, we used double-colored visualizations. In these visualization, yellow color can be considered to correspond to "important" and red color to "very important" parts. Below, we give the details of our crowdsourcing task design.

Task Design

The main challenge of our crowdsourcing task is the non-familiarity of the data. Thus, in the task design, we provided supervision to non-experts in two ways. Considering that humans are good at generalizing from few samples, we provided a couple of examples that belong to the same class as the target glyph. As illustrated in the top parts of Fig. 6.18 and 6.19, in these examples, we also marked their *possible* diagnostic parts. Furthermore, as we hypothesize that people are good at relative analysis, we provided examples from other classes as well. In the task instructions (Fig. 6.17-6.18), we provided an example from the 50 classes used in training the CNN models. In the instructions, we also provided detailed explanation of the ratings and what they visually correspond to (Fig. 6.18). The task itself is composed of two parts. In the first part, the annotator observes a visualization of the CNN output and rates it in a scale of 7 ranging from "very poor" to "very good". We ask the annotator to rate each of the networks' visualizations separately. In the second part, we show the visualizations from the two networks on the same glyph image, and ask the annotator to rate them relative to each other.

Using the Crowdflower terminology introduced in Chapter 4.4.1, we set 4 tasks in a page, and paid an annotator 10 dollar cents per page. We collected annotations from 10 annotators per task. We set 10 test questions to be used in quiz mode. Quiz mode enables to eliminate spammers or low-performing annotators. In total, we collected 1000 annotations (10 for each of the 100 glyphs).

Table 6.2 – Aggregated results of interpretability crowdsourcing analysis on the pilot set for the first 5 classes. Blue frames indicate the crowdworkers' preference in relative ratings.

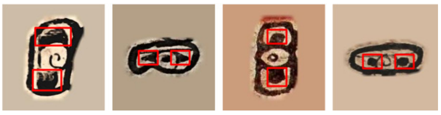
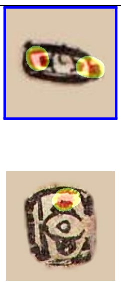
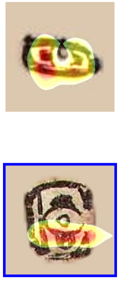
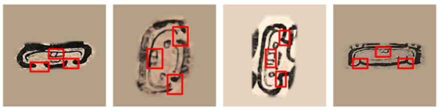
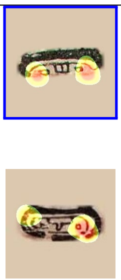
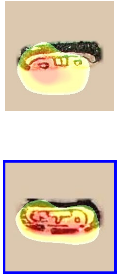
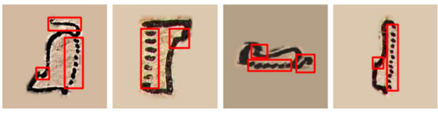
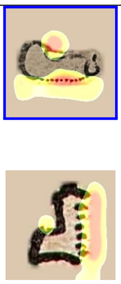
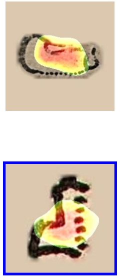
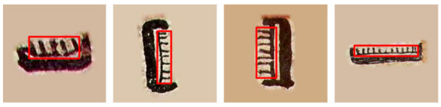
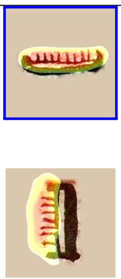
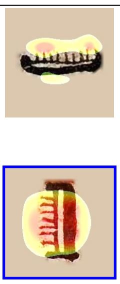
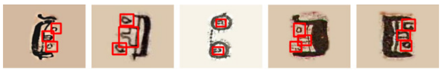
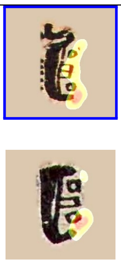
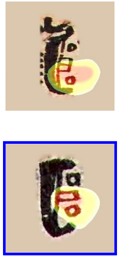
Class	**Diagnostic Part**	**SaN vis.**	**RN50 vis.**	**SaN did better (over 10 glyphs)**	**RN50 did better (over 10 glyphs)**
1B1	 Filled (black) markings on two sides of the central circle			8	1
1G1	 Two discontinuities at the bottom of outer contour, "teeth" attached to the inner thin line			1	9
1S2	 Dots at the bottom of the glyph, dented top part of the glyph, and a curvy end if exists			7	3
AA1	 A series of parallel elongated lines			8	1
HE6	 Two hollow circles, hollow "teeth" attached to the inner thin line (in case of thick outer frame)			8	1

Table 6.3 – Aggregated results of interpretability crowdsourcing analysis on the pilot set (continued for the rest of the 10 classes). Blue frames indicate the crowdworkers' preference in relative ratings.

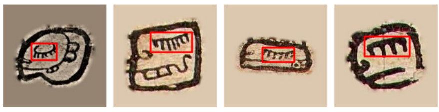
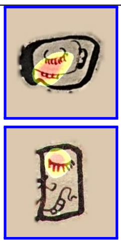
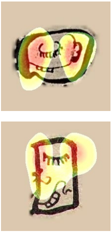
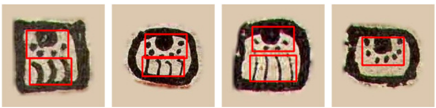
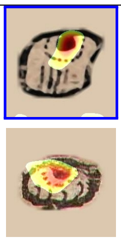
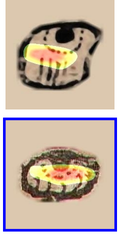
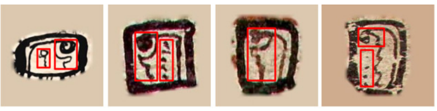
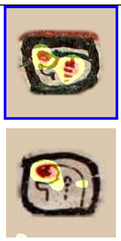
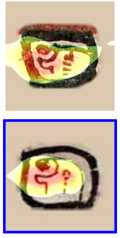
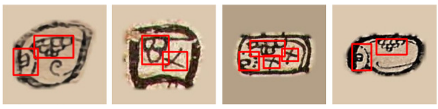
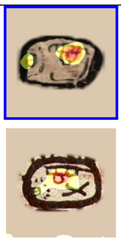
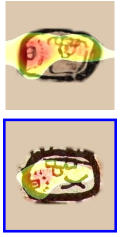
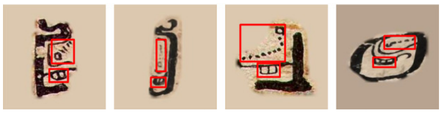
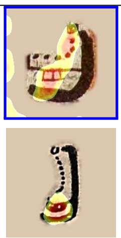
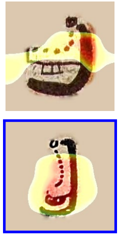
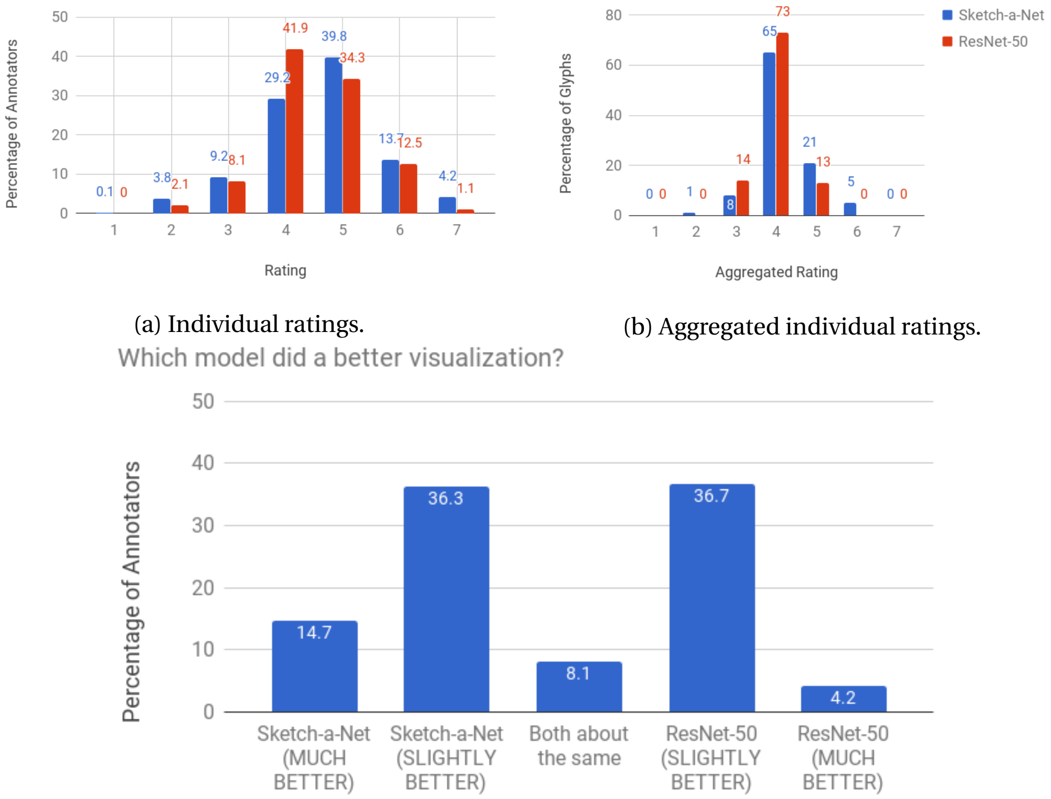
Class	**Diagnostic Part**	**SaN vis.**	**RN50 vis.**	**SaN did better (over 10 glyphs)**	**RN50 did better (over 10 glyphs)**
SCC	 Closed "eye" with "eyelashes"			9	0
XE2	 Filled (black) full circle at the inner top, uniform-sized dots surrounding the filled circle, parallel vertical lines attached to the bottom of glyph			7	3
YS1	 Inner filled (black) circle with a curvy "tail", inner consecutive dots			3	6
ZC1	 "Grape"-like set of small inner circles, one or more "x" signs, hollow "teeth" with half-circular dots around them			3	7
ZU1	 One hollow circle with following dots, hollow "teeth" attached to the inner thin line			4	6

Table 6.4 – Some of the comments from the crowdworkers on the interpretability of the two CNN visualizations (Sketch-a-Net -SaN- vs. ResNet50 -RN-) about the glyph diagnostic parts. In the comments, Robot A and B refer to the SaN and RN models, respectively. In the relative ratings, 0, 1, or 2 indicates ”about the same,“ ”slightly better,“ ”much better,“ respectively.

Class	Crowdworker Comment	Ind. rating (SaN)	Ind. rating (RN)	Rel. rating
1B1 (top in Table 6.2)	Robot A was more specific in highlighting the most significant parts of the glyph, while robot B highlighted most of the glyph	6	4	SaN_2
AA1	A seems to be a bit over the area, but compared to B it's much better.	6	3	SaN_2
AA1	A is perfect, covers the background a bit, but it's perfect.	7	6	SaN_2
SCC	Robot A almost had a full match but also marked a little background. Robot B had a full match but it also marked too much background.	6	5	SaN_2
XE2	The A focuses on a single point of the figure but still is better than the B	5	4	SaN_1
XE2	Robot A was confident most of one feature of the glyph, while Robot B misses most (not all) of the same feature.	3	2	SaN_2
ZC1	The robot B emphasized more precise parts than the robot A	3	6	RN_1
ZU1	B is slightly better as it covers more of the diagnostic areas whereas A is only about half of B.	3	5	RN_1
ZU1	B it's just a random mark in the middle, A seems to be more accurate.	5	3	SaN_1
ZU1	Robot A: Confident only at the edge of diagnostic part. Robot B: Highlighted background and other parts mostly	6	4	SaN_2
ZU1 (top in Table 6.3)	Robot A was almost perfect while the B one hardly touched the diagnostic parts .	6	4	SaN_2
1S2	Robot A had more than a half of the match and too much background. Robot B had a little match and also matched a little background.	5	4	SaN_1
1G1 (top in Table 6.2)	Robot A missed one important part, and robot B marked big area of unimportant part, so it is equal.	5	5	0
1G1	Robot B confident only at the edge of diagnostic part, Robot A missed some of diagnostic part	5	7	RN_1
1G1	Figure A, although not so much encompassing, focuses on the crucial points of the form	5	4	SaN_1
1B1	Robot B was confident about the two features of the glyph, while Robot A was confident about only one.	5	6	RN_2



(a) Individual ratings.

(b) Aggregated individual ratings.

Which model did a better visualization?

(c) Relative ratings.

Figure 6.20 – The distributions of (a) individual ratings, (b) individual ratings aggregated per glyph, and (c) relative ratings in the pilot crowdsourcing study on interpretability of CNN representations.

Analysis of Crowdsourced Ratings

To understand the preference of crowdworkers among the visualizations of the studied two CNNs, we, first, analyzed the distributions of individual ratings (shown in Fig. 6.20a) with the Kolmogorov-Smirnov (KS) 2-sample test Massey Jr [1951]. KS 2-sample test rejects the null hypothesis that samples come from the same distribution based on comparing the p-value with the significance level α [Massey Jr, 1951]. In our case, KS test rejected the null hypothesis at significance level $\alpha = 0.005$. This result suggested that the crowdworkers did not perceive the heatmaps from the two CNN models as the same.

Secondly, we analyzed the aggregated individual ratings of the glyph instances. To obtain these aggregated ratings, we simply averaged all the individual ratings from 10 annotators that were assigned to the annotation of that glyph. Complementary to the point above, in Fig. 6.20b, we observed that the percentage of the glyphs that is rated as positive (rating 5, 6 or 7) in the overall aggregated ratings are higher for the Sketch-a-Net visualizations ($21+5=26$ %) than for the Residual Network visualizations (13 %).

Thirdly, to check the relative ratings, we aggregated the relative ratings of 10 annotators per glyph in a weighted manner. The “much better” options (extreme end of the scale) get 2 votes whereas “slightly better” options get 1 vote. Overall, among the 100-glyphs, the crowdworkers favored the Sketch-a-Net visualizations over the ResNet-50 visualizations for 52 glyphs. ResNet-50 visualizations were favored for 41 glyphs. There was a tie for 7 glyphs. The last two columns of Table 6.2-6.3 show these distributions across classes. From these class-based results, we observed that the ResNet-50 visualizations were found more appealing for the categories with several diagnostic parts (i.e. ZC1, YS1, ZU1, and 1G1). In these categories, the diagnostic parts cover almost the whole glyph. Since the ResNet-50 heatmaps, in general, highlight more regions than the Sketch-a-Net heatmaps, we considered this finding plausible.

Finally, in Table 6.4, we listed a few of the insightful comments from the crowdworkers. These comments help to understand what kind of rating criteria were used by the crowdworkers.

6.4 Conclusion

In this chapter, first, we explored visualization of glyph shape representations via t-SNE over various Maya corpora. t-SNE is employed as a non-linear dimensionality reduction method in order to map the high-dimensional visual representations to 2-D coordinate space. By inspecting the visual clusters obtained via t-SNE, we observed that glyphs with different contour styles (i.e. thin vs. thick, rectangular vs. circular, or composed of discontinuous dots), distinct compositions (composed of one or several elements), and different complexity of inner details are grouped separately. These visualizations illustrate that the studied shape representations and the dimensionality reduction method are promising and could be used as an effective way to observe the visual structure in the shape datasets.

Secondly, we visualized the discriminative parts of glyphs via guided gradient backpropagation and Grad-CAM methods, and showed that the trained model has a great potential as the discriminative parts of glyphs matched with the expert descriptions in a 5-glyph case study. To our knowledge, this is the first time that expert knowledge in ancient Maya epigraphy is reflected in a fully data-driven machine inference process, i.e. that does not use hand-crafted shape descriptors. Additionally, we showed the potential of the Grad-CAM method in glyph localization in a cluttered setting, i.e. glyph-blocks. Finally, we investigated on how to assess and exploit CNN visual outputs in a comparative study. According to this study, sketch-specific network (SaN) visualizations were found more focused and appealing compared to the more-diffused visualizations of the residual network (ResNet-50). The ResNet-50 visualizations were favored only in the case of the glyph categories that require a large region to diagnose. Therefore, we conclude that overall the crowdworkers perceived the SaN visualizations more precise and insightful than the ResNet-50 visualizations in order to locate the diagnostic parts of the glyphs.

7 Conclusions and Perspectives

In this thesis, we investigated Social Computing and Computer Vision methods to provide computational support to scholars in Digital Humanities, in the context of analyzing cultural heritage materials from the ancient Maya civilization. Specifically, our first goal was to explore a crowdsourcing approach as an option in time-consuming expert tasks. Our second goal was to examine shallow and deep data-driven visual representations as alternative to traditional visual descriptors for supervised classification. This chapter concludes the thesis by listing the contributions of the individual chapters, and by discussing the limitations of our work as well as possible future research directions.

7.1 Contributions

Local Shape Representations. In Chapter 3, we assessed a knowledge-driven shape descriptor (HOOSC) and a data-driven visual representation (a single-layer Sparse Autoencoder) on the syllabic monument glyph data and a sketch benchmark dataset. We drew the conclusion that relatively large datasets are needed for data-driven approaches to outperform traditional visual representations. Furthermore, we decided to use deeper architectures in the next chapters to capture the regularities in the data.

Crowdsourcing. In Chapter 4, we designed two crowdsourcing studies. The first study showed that, with minimal supervision (indicating the number of glyphs existent in the block), non-experts can mark the glyph boundaries correctly in simple cases such as 3 glyphs in a block. We also learned that providing supervision is key in crowdsourcing tasks that crowdworkers are neither familiar with the data nor the task. Based on this outcome, in our second study, we re-designed the glyph segmentation task carefully with more supervision (presenting the variants for the glyph of interest). We conducted three gradual stages to control several task parameters. First, in the preliminary stage, we discovered that a thoroughly simplified design is essential. Apart from the design, allowing only the high-performing crowdworkers, and setting the payment to a decent value also helped to obtain high-quality outcomes. Secondly, in the small-scale stage, we evaluated the crowd segmentation performance over 800 glyphs from

134 categories.In this stage, we concluded that non-experts were able to generate high-quality segmentations with the simplified and improved design. We also pointed out that, according to the task difficulty and similarity ratings of the crowd, the glyph variants from the modern Macri and Vail [2008] catalog provided better supervision than the variants in the [Thompson and Stuart, 1962] catalog. Therefore, we proceeded to the next stage with the Macri-Vail glyph variants. Finally, in the large-scale stage, we obtained valid segmentations for over 9000 glyphs. 8661 glyphs belong to the most frequent 150 classes in our dataset. This is a unique resource for future research.

Visual Representations with Deep CNNs. In Chapter 5, to address how to recognize complex shapes with small amount of data, we studied two traditional shape descriptors and three training approaches with CNNs over a challenging Maya codical glyph dataset. Specifically, we assessed HOG and HOOSC descriptors, representations learned in several existing pretrained networks, i.e. Sketch-a-Net, VGG-16, and ResNet-50, finetuning the last blocks of these pretrained nets, and training the sequential and residual CNN variants from scratch. We showed that pretrained CNN representations outperform traditional descriptors by a large margin. Furthermore, fine-tuning the last convolutional layer improved the classification performances considerably, compared to the pretrained representations. We observed that the VGG-16 pretrained network is more robust than the recent ResNet-50 network for assessing the data with different nature (as is the case of glyphs). At this point, we can conclude that the transfer learning approach we studied (i.e. via fine-tuning the last convolutional block of a deep CNN pretrained on a large-scale dataset) can be applied to other tasks such as handwriting recognition even in the case of small amount of data. That said, training a sequential sketch-specific network with few parameters from scratch with batch normalization (eliminating the need of pre-training in the early layers of CNN), balanced oversampling, and dropout regularization outperformed the other training strategies and the recent residual models. Note that this model achieved over 70% average top-1, and over 85% average top-5 accuracy in the 150-class case, and proved itself to be useful in a retrieval scenario. This finding is rather promising for all the other visual shape recognition tasks with limited amount of data.

Glyph Visualization. In Chapter 6, first we studied visualization methods of shape representations via t-SNE over various Maya corpora. We showed that this is an effective way to observe the structure in the data. Secondly, we visualized the discriminative parts of glyphs via Grad-CAM method, and showed that the trained CNN models can highlight the discriminative parts of glyphs where the experts consider as diagnostic. Additionally, we showed the potential of the Grad-CAM method in glyph localization in a cluttered setting, i.e. glyph-blocks. Finally, we investigated on how to assess and exploit CNN visual outputs in a comparative crowdsourcing study. According to this study, sketch-specific network visualizations were found more interpretable and more often more appealing to the crowd compared to the more-diffused residual network visualizations.

7.2 Limitations and Perspectives

This dissertation focused on visual analysis and crowdsourcing approaches for Maya writings. Below, we discuss the limitations of our methods and possible future directions that are structured under five main points.

Crowdsourcing. We investigated crowdsourcing as an alternative to expert tasks. As a possible future crowdsourcing approach, as an alternative to our static glyph segmentation design, interactive task design may be preferred to prompt non-experts to produce higher-quality outcomes. The envisioned feedback mechanism in an interactive design might involve online classification of regions that are marked by crowdworkers. This way, crowdworkers may be warned in case of mismatch of glyph label and target region.

Improving Visual Representation Learning. To improve visual representation learning, we can consider exploring more recent deep neural network architectures (such as the variants of densely-connected convolutional networks (DenseNet) [Huang et al., 2016]) that can learn more competitive representations without dramatic increase of network parameters. As we tackle recognition with limited amount of original data, we are interested in efficient representations that can be achieved with small number of parameters. In this line, Ha et al. [2016] proposed to use an auxiliary network (called as “hypernetworks”) to learn the structure of the main networks’ parameters during training. This approach is discussed to be applicable to a large variety of network architectures, and, in an efficient use of parameters, this approach produces small-sized representations with comparable performances to the original networks.

Furthermore, we can employ the variants of promising recent techniques while training deep neural networks, for instance, dense-sparse-dense regularization [Han et al., 2016], knowledge distillation [Hinton et al., 2015], and snapshot ensembles [Huang et al., 2017], i.e. capturing snapshots of a model during training and ensembling these models for obtaining the final result. As the ensemble of networks outperforms individual networks in general (e.g. [Simonyan and Zisserman, 2014]), we can also consider to merge the decisions from the individual networks that we have trained on the different splits of the data in Chapter 5.

Another possible line of research could be on metric learning. Schroff et al. [2015] proposed incorporating “triplet loss” at the end of a deep CNN architecture. This loss minimizes the distance between two samples from the same class, while maximizing their distances to a sample from a different class in the predefined “triplets” of the data samples. This kind of approach may be useful to handle our unbalanced limited original glyph data in the classification tasks, especially in the case of very few (less than five) samples per class. Similar to the work by Jose and Fleuret [2016], another metric learning approach could be combining and compressing various glyph representations from different networks in a joint optimization scheme. This way, we might be able to learn more compact representations that may bring together the merits of the pretrained and the glyph-specific representations.

In this thesis, we experimented in supervised setting, and another line of possible research could be on unsupervised learning. In this way, we can explore glyph-block images that have not been used in our experiments. For instance, Generative Adversarial Networks [Denton et al., 2015; Goodfellow et al., 2014; Radford et al., 2015] can be considered for both learning shape representations (e.g. as in [Donahue et al., 2016]) and for restoration of degraded parts, since these methods were reported to be competitive to capture the regularities in the data in case of large-scale unlabeled data along a small-scale labeled dataset.

Incorporating Textual Information. This thesis focused on the visual analysis of Maya glyphs. In a complementary angle, textual information (i.e. glyph category, transcription and translation), metadata (i.e. relative location of glyphs and semantic context of containing pages in codices), and context (i.e. co-occurrence frequencies of glyphs) could be considered to be a basis of automatic analysis tools for scholars. As shown by Hu et al. [2014], in a glyph retrieval task, incorporating context (co-occurrence frequencies) along visual representation improves the average ranking results significantly. Aligned with this insight, we hypothesize that incorporating other extra information could help recognition especially in the codices case. The ordered glyphs in the codices could be used to train sequential models such as Recurrent Neural Networks (RNNs) [Graves et al., 2009, 2012] or Long-Short Term Memory (LSTM) networks [Hochreiter and Schmidhuber, 1997]. These networks might be used to predict the category of a damaged glyph based on the remaining visual content and the neighboring glyph statistics. A step towards the machine translation of Maya text could be making use of the possible transcription and translation sources (e.g. as in the Maya Codices webpage of Vail and Hernandez [2013]) along with the visual content in a multi-task sequence learning setting as discussed by Luong et al. [2015].

Reconstructing Degraded Glyphs. As described in Chapter 2, Maya texts may suffer from degradation over time or due to external factors. Another interesting future task could be to refine the degraded glyphs in the glyph-blocks. This refinement may include denoising the glyph regions, sharpening blurry contours, refilling small holes and reconstructing partially missing parts. It could be interesting to address denoising, sharpening, and refilling holes in a generic pipeline, or to transfer the knowledge from one task to another, as discussed by Xiao et al. [2017]. However, we hypothesize that reconstructing a partially missing part may require a different strategy. For this task, generative sequential patch-based models might be explored. Recently, Variational Autoencoders (VAE) [Kingma and Welling, 2013] and their variants with an attention mechanism (i.e. DRAW [Gregor et al., 2015, 2016]), Generative Adversarial Networks (GANs) [Denton et al., 2015; Goodfellow et al., 2014; Radford et al., 2015], and auto-context based models (i.e. PixelCNN and PixelRNN [van den Oord et al., 2016a,b]) have been shown to generate realistic data samples. We hypothesize that these methods could be investigated for glyph image completion tasks.

Automatic Glyph Segmentation. Thanks to the crowdsourced individual glyph dataset generated and described in Chapter 4, we were able to train models for a large variety of glyph categories in Chapter 5. From this point on, we can consider using these models as basis for automatic localization or segmentation for the unexplored glyphs in our crowdsourcing study. Still, for this task, we hypothesize that using a model with few parameters is essential. Therefore, a variation of DenseNet that is adapted to semantic segmentation may be considered as in [Jégou et al., 2016].

Among recent semantic image segmentation approaches, there have been promising approaches that combine Conditional Random Fields (CRF) and deep neural networks (either CNNs or RNNs) [Lin et al., 2015; Zheng et al., 2015]. For instance, Lin et al. [2015] proposed training a piecewise CRF on top of the multi-scale Fully Convolutional Neural Networks (FCN) Long et al. [2015]. The authors remark that they compute both the unary and pairwise potentials via FCNs, and efficient joint piecewise training makes the inference computationally feasible.

Furthermore, weakly-supervised pixel-level segmentation is also relevant to our case, in the sense that our Codices glyph-block corpus have glyph annotations which correspond to image-level object class labels. Along this line, Pinheiro and Collobert [2015] propose to formulate this problem as a multiple instance learning task. Similarly Oquab et al. [2015] proposes a weakly-supervised approach with CNNs that outputs approximate locations (without the extent of the objects). This method performs on par with a recent fully-supervised method [Girshick et al., 2014]. Aligned with these approaches, we hypothesize that the Grad-CAM visualization (used in Chapter 6) can be also a starting point towards weakly-supervised glyph localization and segmentation.

Keeping these computational approaches in mind, we consider that there are various opportunities in the future to enhance the available computational support to Digital Humanities scholars.

A Exploring HOOSC Shape Descriptor

In this appendix chapter, we present the study on the settings of the traditional HOOSC shape descriptor for individual syllabic glyph classification. This study is previously published as the Section *Shape-based Glyph Classification* in the following journal paper:

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

A.1 Methodology

The objective is to build a classifier that categorizes a test shape into one of the N_G categories. Given a test data G and another glyph in the dataset D represented by histograms H^G and H^D, we compute the Cityblock distance to measure the dissimilarity between G and D:

$$d(G,D) = \sum_{1\leq i\leq k} |H^G(i) - H^D(i)|. \tag{A.1}$$

where each histogram is normalized so that $\sum_{1\leq i\leq k} H(i) = 1$.

As a baseline, we use the method of [Roman-Rangel et al., 2011b], where glyphs are represented using the global bag-of-words (BoW) representation [Sivic and Zisserman, 2003a]. In this classification case, a test glyph get the class label of its nearest neighbor (using the BoW cityblock distance of Eq. A.1) in the training set.

As an alternative, we propose a method that categories an unknown glyph by first identifying the category of its individual local pivot points. Specifically, for a given glyph, we first compute

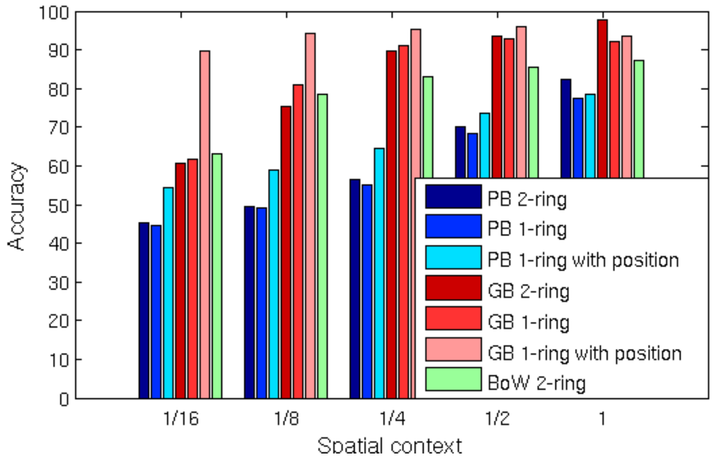


Figure A.1 – Classification accuracy of the BoW method (green bar), and of the proposed method (at the pivot level -pivot-based 'PB' results, blue bars- and at the glyph level -glyph-based results, 'GB', red bars; see VI.A for details about the methods), for different spatial context sizes and settings to compute the HOOSC descriptor. See VI.B.2) and 3) for more details.

the HOOSC descriptor at each pivot point and classify it using a K Nearest Neighbor method. Then, in a second step we classify the glyph as the category that receives the largest amounts of votes from the individual pivots.

A.2 Experimental Results

A.2.1 Dataset

We used a subset of glyphs from monumental inscriptions that were exploited in [Roman-Rangel et al., 2011b]. This dataset is introduced in the Section 2.4 in the Chapter 2. We only consider the glyph categories which has more than 30 glyphs. The resulted dataset is composed of 10 glyph categories with 25 training images per class and 125 test images in total. The groundtruth of the glyph category is provided by our team scholars.

A.2.2 Experimental Setting

We used 300 equidistant pivots where we compute the shape descriptor. Note that here, we extracted the orientation from the raw images preprocessed by a continuous Gaussian orientation filter, as this gave more stable results than applying the thinning pre-processing.

We considered three settings to compute the HOOSC descriptor: (1) HOOSC with 2 rings and 8 radial bins; (2) HOOSC with 1 ring and 8 radial bins, see Fig. A.2; (3) case (2) with position, i.e. where the HOOSC descriptor is augmented with the relative position (defined within $[0;1] \times [0;1]$) of the pivot point within the glyph bounding box.

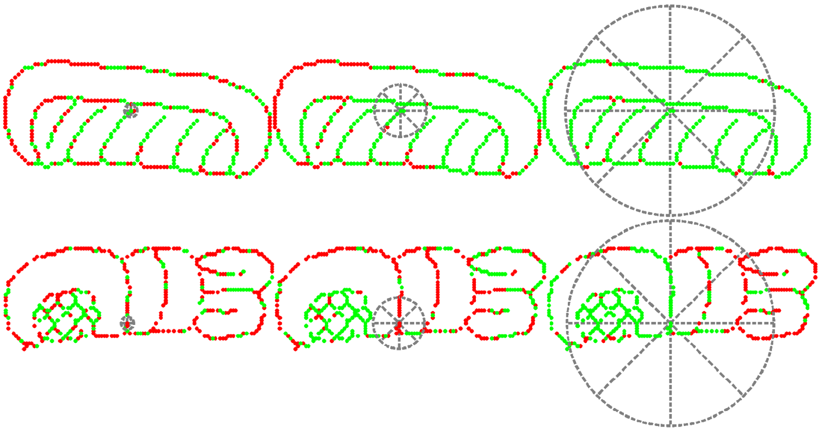


Figure A.2 – Classified pivots using HOOSC 1-ring with position and spatial context 1/16 (left), 1/4 (middle) and 1 (right) respectively. Green (red) points indicates pivot correctly (uncorrectly) classified with the glyph class label.

Furthermore, for each of the three settings, we considered five spatial context (radius of the outer ring in HOOSC computation): 1/16, 1/8, 1/4, 1/2, and 1, all defined as a proportion to the mean of the pairwise distance between pivot points (see gray circles in Fig. A.2), as we are interested in studying the impact of the spatial scope used to compute the HOOSC descriptor on the classification performance. Indeed, while large scopes used in previous works (and the retrieval Section) led to good results when dealing with clean glyph inputs, there are situations where smaller scopes would be useful, like when dealing with damaged glyph shapes (the damage will affect most of the descriptors when using a large scope), or if we want to identify which local part of the glyph is a 'diagnostic' feature, i.e. a discriminant feature that scholars rely on to distinguish a glyph.

A.2.3 Results and Discussion

Fig. A.1 shows the classification results obtained using the BoW method and the proposed method ('glyph-based' results, denoted GB) for different spatial context sizes and partition settings. In order to better understand the proposed method, we also show the 'pivot-based' (denoted PB) classification accuracy, i.e. the percentage of pivot points whose descriptor is correctly labeled with the class of its associated glyph.

First, from the results of the 'pivot-based' method (blueish bars), we can notice that the performance decreases almost linearly as the spatial context becomes smaller, but remains well above chance level (10%) even for the very small spatial extent (1/16). Interestingly, as this context gets smaller, the incorporation of the spatial position (PB 1-ring with position) allows to boost performance by 10% as compared to without position (PB 1-ring). Furthermore, while 2 rings are useful as the spatial context is large (e.g. 1), it is not superior than 1 ring in terms of PB performance and actually degrades the GB performance when smaller spatial context is considered (e.g. 1/4 to 1/16).

Secondly, the performance w.r.t. spatial context at the glyph level (GB results, reddish bars) does not decrease as dramatically than at the pivot level, indicating that misclassified points, even if they dominate, usually get distributed over all other classes rather than a single one. Hence the pivots predicted with true labels may win in the voting phase. For the GB 1-ring with position, the classification remains as high as 94% with a spatial context of 1/8. Note that this is not the case of the BoW approach (green bars), whose performance regularly degrades as the spatial context decreases, performing worse than the proposed approach with spatial radius larger than 1/4, and can not keep up with the 1-ring with position results at smaller spatial scopes.

Fig.A.2 illustrates pivot classification result for two glyphs over three spatial context levels. We can see that the number of pivots classified correctly increases with the spatial context. It also shows that while some local structures are recognized at most scales (individual diagonal lines for the top glyph, hatches in the bottom one), there are structure that still remain confusing with other glyphs, even at the larger contexts (pivot near the 'ears' in the bottom glyph).

We can conclude that a two step approach where class-information is used at categorizing the descriptor (rather than simply quantization in BoW) brings more robustness as the spatial context decreases (and may bring even more robustness when dealing with partially damaged glyphs) and that incorporating the relative position of pivots is important, as the same local shape structure might be observed at different positions for different glyph categories.

Bibliography

Abrar H Abdulnabi, Gang Wang, Jiwen Lu, and Kui Jia. Multi-task cnn model for attribute prediction. *IEEE Transactions on Multimedia*, 17(11):1949–1959, 2015.

Sebastiano Battiato, Giovanni Maria Farinella, Giovanni Gallo, and Daniele Ravì. Exploiting textons distributions on spatial hierarchy for scene classification. *Journal on Image and Video Processing*, 2010:7, 2010.

Sebastiano Battiato, Giovanni Maria Farinella, Oliver Giudice, and Giovanni Puglisi. Aligning shapes for symbol classification and retrieval. *Multimedia Tools and Applications*, pages 1–19, 2015.

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, pages 509–522, 2002.

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

Yoshua Bengio and James S. Bergstra. Slow, decorrelated features for pretraining complex cell-like networks. In *Conference on Advances in Neural Information Processing Systems 22*, pages 99–107, 2009.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8): 1798–1828, 2013.

Paris Bibliotheque Nationale. Paris Codex. http://gallica.bnf.fr/ark:/12148/btv1b8446947j. Improved reproduction of the Léon de Rosny color edition of Paris 1887, by courtesy of Akademische Druck - u. Verlagsanstalt - Graz, Austria.

J. I. Biel and D. Gatica-Perez. The youtube lens: Crowdsourced personality impressions and audiovisual analysis of vlogs. *IEEE Transactions on Multimedia*, 15(1):41–55, Jan 2013. ISSN 1520-9210. doi: 10.1109/TMM.2012.2225032.

Chiara Bonacchi, Andrew Bevan, Daniel Pett, Adi Keinan-Schoonbaert, Rachael Sparks, Jennifer Wexler, and Neil Wilkin. Crowd-sourced archaeological research: The micropasts project. *Archaeology International*, 17, 2014.

Y-L Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 2559–2566. IEEE, 2010.

Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *ECCV*, pages 438–451. Springer, 2010.

Ernest Alfred Wallis Budge. *The Book of the Dead: an English translation of the chapters, hymns, etc., of the Theban recension, with introduction, notes, etc.*, volume 6. Open Court Pub., 1901.

Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Is that a jaguar? Segmenting ancient Maya glyphs via crowdsourcing. In *ACM International Workshop on Crowdsourcing for Multimedia*, pages 37–40. ACM New York, November 2014. doi: 10.1145/2660114.2660117.

Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Evaluating shape representations for Maya glyph classification. *ACM Journal on Computing and Cultural Heritage*, 9(3), Sep 2016a.

Gulcan Can, Jean-Marc Odobez, Carlos Pallan Gayol, and Daniel Gatica-Perez. Ancient Maya writings as high-dimensional data: a visualization approach. In *Digital Humanities*, 2016b.

Gülcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Shape representations for maya codical glyphs: Knowledge-driven or deep? In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, CBMI '17, pages 32:1–32:6, New York, NY, USA, 2017a. ACM. ISBN 978-1-4503-5333-5. doi: 10.1145/3095713.3095746. URL http://bib-ezproxy.epfl.ch:2512/10.1145/3095713.3095746.

Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. How to tell ancient signs apart? Recognizing Maya glyphs with CNNs. Idiap-RR Idiap-Internal-RR-26-2017, Idiap, April 2017b.

Gulcan Can, Jean-Marc Odobez, and Daniel Gatica-Perez. Maya codical glyph segmentation: A crowdsourcing approach. Research Report Idiap-RR-01-2017, Idiap, January 2017c.

Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

L Carletti, G Giannachi, and D McAuley. Digital humanities and crowdsourcing: An exploration. *Museums and the Web, Portland, Oregon.*, 2013a.

Laura Carletti, Gabriella Giannachi, Dominic Price, and Derek McAuley. Digital humanities and crowdsourcing: An exploration. In *Museum and the Web*, pages 223–236, 2013b.

Tim Causer and Melissa Terras. "many hands make light work. many hands together make merry work": Transcribe bentham and crowdsourcing manuscript collections. pages 57–88. Ashgate Surey, 2014.

K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*, 2011.

L Chen, F Rottensteiner, and C Heipke. Feature descriptor by convolution and pooling autoencoders. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:31–38, 2015.

Dan C Cireşan, Ueli Meier, and Jürgen Schmidhuber. Transfer learning for latin and chinese characters with deep neural networks. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–6. IEEE, 2012.

Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005a.

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

Charles Etienne Brasseur de Bourbourg. *Relation des choses de Yucatan de Diego de Landa*. Durand, 1864.

A. Del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *PAMI*, pages 121–132, 1997.

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

Emily L Denton, Soumith Chintala, arthur szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1486–1494. Curran Associates, Inc., 2015.

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. In *ICML*, volume 32, pages 647–655, 2014.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. Sketch-based 3D shape retrieval. In *SIGGRAPH Talks*, 2010.

Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph.*, 31(4):44:1–44:10, jul 2012a. ISSN 0730-0301. doi: 10.1145/2185520.2185540.

Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Transactions on Graphics (Proceedings SIGGRAPH)*, 31(4):44:1–44:10, 2012b.

Sergio Escalera, Alicia Fornés, Oriol Pujol, Josep Lladós, and Petia Radeva. Circular blurred shape model for multiclass symbol recognition. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(2):497–506, 2011.

E.B. Evrenov, Y. Kosarev, and B.A. Ustinov. *The Application of Electronic Computers in Research of the Ancient Maya Writing.* USSR, Novosibirsk, 1961.

Orhan Firat. liborf: A machine learning toolkit for deep learning, probabilistic graphical models and structured prediction, 2015. Available from: <http://www.ceng.metu.edu.tr/ e1697481/libORF.html>. Accessed: 2015-06-03.

Andreas Fischer, Emanuel Indermühle, Horst Bunke, Gabriel Viehhauser, and Michael Stolz. Ground truth creation for handwriting recognition in historical documents. In *IAPR International Workshop on Document Analysis Systems*, pages 3–10. ACM, 2010.

Lucy Fortson, Karen Masters, and Robert Nichol. Galaxy zoo. *Advances in machine learning and data mining for astronomy*, 2012:213–236, 2012.

Morris Franken and Jan C. van Gemert. Automatic Egyptian hieroglyph recognition by retrieving images as texts. In *ACM Multimedia Conference*, pages 765–768, 2013.

Alan Henderson Gardiner. *Egyptian grammar: being an introduction to the study of hieroglyphs.* Published on behalf of the Griffith Institute, Ashmolean Museum, Oxford, by Oxford University Press, 1957.

Basilis Gatos, Georgios Louloudis, Tim Causer, Kris Grint, Verónica Romero, Joan Andreu Sánchez, Alejandro H Toselli, and Enrique Vidal. Ground-truth production in the transcriptorium project. In *IAPR International Workshop on Document Analysis Systems*, pages 237–241. IEEE, 2014.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

L. Gottlieb, G. Friedland, J. Choi, P. Kelm, and T. Sikora. Creating experts from the crowd: Techniques for finding workers for difficult tasks. *IEEE Transactions on Multimedia*, 16(7): 2075–2079, Nov 2014. ISSN 1520-9210. doi: 10.1109/TMM.2014.2347268.

Ian Graham. *Corpus of Maya hieroglyphic inscriptions*, volume 3. Peabody Museum of Archaeology and Ethnology, Harvard University, 1979.

Ian Graham and Eric Von Euw. *Corpus of Maya hieroglyphic inscriptions*, volume 3. Peabody Museum of Archaeology and Ethnology, Harvard University, 1977.

Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2009.

Alex Graves et al. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer, 2012.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra. Towards conceptual compression. In *Advances In Neural Information Processing Systems*, pages 3549–3557, 2016.

Danna Gurari, Diane Theriault, Mehrnoosh Sameki, and Margrit Betke. How to use level set methods to accurately find boundaries of cells in biomedical images? evaluation of six methods paired with automated and crowdsourced initial contours. In *MICCAI: Interactive Medical Image Computation (IMIC) Workshop*, page 9, 2014.

Danna Gurari, Diane Theriault, Mehrnoosh Sameki, Brett Isenberg, Tuan A Pham, Alberto Purwada, Patricia Solski, Matthew Walker, Chentian Zhang, Joyce Y Wong, et al. How to collect segmentations for biomedical images? a benchmark evaluating the performance of experts, crowdsourced non-experts, and algorithms. In *Winter Conf. on Applications of Computer Vision*, pages 1169–1176. IEEE, 2015.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Song Han, Jeff Pool, Sharan Narang, Huizi Mao, Shijian Tang, Erich Elsen, Bryan Catanzaro, John Tran, and William J. Dally. DSD: regularizing deep neural networks with dense-sparse-dense training flow. *CoRR*, abs/1607.04381, 2016. URL http://arxiv.org/abs/1607.04381.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Paulina Hensman and David Masko. The impact of imbalanced training data for convolutional neural networks. Technical report, KTH, Stockholm, Sweeden, 2015. Degree Project, in Computer Science, First Level.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Geoffrey E. Hinton and Sam T. Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 833–840, 2002.

Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length, and helmholtz free energy. *Conference on Advances in Neural Information Processing Systems*, pages 3–3, 1994.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

Judy Hoffman, Eric Tzeng, Jeff Donahue, Yangqing Jia, Kate Saenko, and Trevor Darrell. One-shot adaptation of supervised deep convolutional models. *arXiv preprint arXiv:1312.6204*, 2013.

Stephen Houston, John Robertson, and David Stuart. The language of classic Maya inscriptions. *Current Anthropology*, 41(3):321–356, 2000. doi: 10.1086/300142.

R. Hu and J. P. Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *CVIU*, pages 790–806, 2013.

Rui Hu, Carlos Pallan Gayol, Guido Krempel, Jean-Marc Odobez, and Daniel Gatica-Perez. Automatic maya hieroglyph retrieval using shape and context information. In *ACM Multimedia Conference*, MM '14, pages 1037–1040, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3063-3. doi: 10.1145/2647868.2655044. URL http://doi.acm.org/10.1145/2647868.2655044.

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

Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of International Conference on Machine Learning*, pages 448–456, 2015.

Humayun Irshad, Laleh Montaser-Kouhsari, Gail Waltz, Octavian Bucur, JA Nowak, Fei Dong, Nicholas W Knoblauch, and Andrew H Beck. Crowdsourcing image annotation for nucleus detection and segmentation in computational pathology: evaluating experts, automated methods, and the crowd. In *Pacific Symposium on Biocomputing*, page 294. NIH, 2015.

Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. *arXiv preprint arXiv:1611.09326*, 2016.

Yangqing Jia, Chang Huang, and Trevor Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *Conference on Computer Vision and Pattern Recognition*, pages 3370–3377. IEEE, 2012.

Cijo Jose and François Fleuret. Scalable metric learning via weighted approximate rank component analysis. In *European Conference on Computer Vision*, pages 875–890. Springer, 2016.

IK. Kazmi, Lihua You, and Jian Jun Zhang. A survey of 2d and 3d shape descriptors. In *Computer Graphics, Imaging and Visualization (CGIV), 2013 10th International Conference*, pages 1–10, Aug 2013. doi: 10.1109/CGIV.2013.11.

Harri J Kettunen and Christophe Helmke. *Introduction to Maya Hieroglyphs: Workshop Handbook*. 2008.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing, 2015. Available from: <http://www.peterkovesi.com/matlabfns/>. Accessed: 2015-01-16.

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

Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

M. Larson, M. Soleymani, M. Eskevich, P. Serdyukov, R. Ordelman, and G. Jones. The community and the crowd: Multimedia benchmark dataset development. *MultiMedia, IEEE*, 19(3): 15–23, July 2012.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178. IEEE, 2006.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998a.

Yann LeCun, Corinna Cortes, and Christopher JC Burges. The mnist database of handwritten digits, 1998b.

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

Biao Leng, Shuang Guo, Xiangyang Zhang, and Zhang Xiong. 3d object retrieval with stacked local convolutional autoencoder. *Signal Processing*, 112:119–128, 2015.

Guosheng Lin, Chunhua Shen, Ian Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv preprint arXiv:1504.01013*, 2015.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

Marcus Liwicki and Horst Bunke. Iam-ondb-an on-line english sentence database acquired from handwritten text on a whiteboard. In *ICDAR*, pages 956–961. IEEE, 2005.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

David G Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157. Ieee, 1999.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015.

Martha J. Macri and Matthew George Looper. *The New Catalog of Maya Hieroglyphs: The Classic Period Inscriptions*, volume 1. University of Oklahoma Press, 2003.

Martha J. Macri and Gabrielle Vail. *The New Catalog of Maya Hieroglyphs, vol. 2: The Codical Texts*. University of Oklahoma Press, 2008.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press, 2008.

Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

M.E. Miller and K.A. Taube. *The Gods and Symbols of Ancient Mexico and the Maya: An Illustrated Dictionary of Mesoamerican Religion.* Thames and Hudson, 1993. ISBN 9780500050682. URL https://books.google.ch/books?id=6YGBQgAACAAJ.

G. Mori, S. J. Belongie, and J. Malik. Efficient shape matching using shape contexts. *PAMI*, 27 (11):1832–1837, 2005.

Madrid Museo de América. Madrid Codex. http://www.famsi.org/mayawriting/codices/madrid.html, 1967. True-color facsimile edition of both sections of the hand painted Maya picture book by courtesy of Akademische Druck - u. Verlagsanstalt - Graz, Austria.

Andrew Ng. Sparse autoencoders lecture notes. https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf, 2013. Accessed: 2015-07-30.

Jiquan Ngiam, Adam Coates, Ahbik Lahiri, Bobby Prochnow, Quoc V Le, and Andrew Y Ng. On optimization methods for deep learning. In *International Conference on Machine Learning*, pages 265–272, 2011.

L. S. Nguyen and D. Gatica-Perez. Hirability in the wild: Analysis of online conversational video resumes. *IEEE Transactions on Multimedia*, 18(7):1422–1437, July 2016. ISSN 1520-9210. doi: 10.1109/TMM.2016.2557058.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

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

Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, pages 2751–2758. IEEE, 2012.

Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision*, pages 143–156. Springer, 2010.

Pedro O. Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Marc Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258, 2016.

E. Roman-Rangel, C. Pallan-Gayol, J.M. Odobez, and D. Gatica-Perez. Retrieving ancient Maya glyphs with shape context. In *ICCV workshop on eHeritage and Digital Art Preservation*, pages 988–995, 2009.

E. Roman-Rangel, C. Pallan-Gayol, J.M. Odobez, and D. Gatica-Perez. Analyzing ancient Maya glyph collections with contextual shape descriptors. *IJCV*, pages 101–117, 2011a.

E. Roman-Rangel, C. Pallan-Gayol, J.M. Odobez, and D. Gatica-Perez. Searching the past: an improved shape descriptor to retrieve Maya hieroglyphs. In *ACM MM*, pages 163–172, 2011b.

E. Roman-Rangel, J.M. Odobez, and D. Gatica-Perez. Assessing sparse coding methods for contextual shape indexing of Maya hieroglyphs. *Multimedia*, 7(2):179–192, 2012.

E. Roman-Rangel, J.M. Odobez, and D. Gatica-Perez. Evaluating shape descriptors for detection of Maya hieroglyphs. In *Mexican Conference on Pattern Recognition (MCPR)*, pages 145–154, June 2013.

Edgar Roman-Rangel. *Statistical Shape Descriptors for Ancient Maya Hieroglyphs Analysis*. PhD thesis, École Polytechnique Fédérale de Lausanne, November 2012.

Edgar Roman-Rangel, Gulcan Can, Stephane Marchand-Maillet, Rui Hu, Carlos Pallan Gayol, Guido Krempel, Jakub Spotak, Jean-Marc Odobez, and Daniel Gatica-Perez. Transferring neural representations for low-dimensional indexing of Maya hieroglyphic art. In *ECCV Workshop on Computer Vision for Art Analysis*, October 2016.

S. Rudinac, M. Larson, and A. Hanjalic. Learning crowdsourced user preferences for visual summarization of image collections. *IEEE Transactions on Multimedia*, 15(6):1231–1243, Oct 2013. ISSN 1520-9210. doi: 10.1109/TMM.2013.2261481.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

Linda Schele, Mary E Miller, and Justin Kerr. The blood of kings: dynasty and ritual in Maya art. 1986.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *CVPR Workshops*, June 2014.

Hoo-Chang Shin, Matthew R Orton, David J Collins, Simon J Doran, and Martin O Leach. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1930–1943, 2013.

E. Siahaan, A. Hanjalic, and J. Redi. A reliable methodology to collect ground truth data of image aesthetic appeal. *IEEE Transactions on Multimedia*, 18(7):1338–1350, July 2016. ISSN 1520-9210. doi: 10.1109/TMM.2016.2559942.

K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, ICCV '03, pages 1470–, 2003a. ISBN 0-7695-1950-4. URL http://dl.acm.org/citation.cfm?id=946247.946751.

Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, pages 1470–1477. IEEE, 2003b.

Alexander Sorokin and David Forsyth. Utility data annotation with amazon mechanical turk. *Urbana*, 51(61):820, 2008.

Hao Su, Jia Deng, and Li Fei-Fei. Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017.

N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, May 2016. ISSN 0278-0062.

The Saxon State and Dresden (SLUB) University Library. Dresden Codex. http://digital.slub-dresden.de/werkansicht/dlf/2967/1/, 1975. True-color facsimile edition of the Maya hand painted book by courtesy of Akademische Druck - u. Verlagsanstalt - Graz, Austria.

John Eric Sidney Thompson and George E. Stuart. *A Catalog of Maya Hieroglyphs*. University of Oklahoma Press, 1962.

Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.

Alfred Marston Tozzer. *Landa's Relacion de las Cosas de Yucatan: a translation*. Peabody Museum of American Archaeology and Ethnology, Harvard University, 1941.

Gabrielle Vail and Christine Hernandez. The Maya codices database, version 4.1, 2013. URL http://www.mayacodices.org/. Accessed: 2015-11-01.

Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016a.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016b.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *The Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

Jan C van Gemert, Jan-Mark Geusebroek, Cor J Veenman, and Arnold WM Smeulders. Kernel codebooks for scene categorization. In *European Conference on Computer Vision*, pages 696–709. Springer, 2008.

Luis Von Ahn and Laura Dabbish. Labeling images with a computer game. In *SIGCHI*, pages 319–326. ACM, 2004.

Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. re-captcha: Human-based character recognition via web security measures. *Science*, 321 (5895):1465–1468, 2008.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Conference on Computer Vision and Pattern Recognition*, pages 3360–3367. IEEE, 2010.

Meng Wang, Youbin Chen, and Xingjun Wang. Recognition of handwritten characters in chinese legal amounts by stacked autoencoders. In *International Conference on Pattern Recognition*, pages 3002–3007. IEEE, 2014.

Wikipedia. Diego de Landa — Wikipedia, the free encyclopedia, 2016. [accessed 10-November-2016].

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492. IEEE, 2010.

Lei Xiao, Felix Heide, Wolfgang Heidrich, Bernhard Schölkopf, and Michael Hirsch. Discriminative transfer learning for general image restoration. *arXiv preprint arXiv:1703.09245*, 2017.

Guo-Sen Xie, Xu-Yao Zhang, and Cheng-Lin Liu. Efficient feature coding based on autoencoder network for image classification. In *Asian Conference on Computer Vision*, pages 628–642. Springer, 2015.

Mingqiang Yang, Kidiyo Kpalma, Joseph Ronsin, et al. A survey of shape feature extraction techniques. *Pattern recognition*, pages 43–90, 2008.

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

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

Matthew D Zeiler. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

Liang Zheng, Yi Yang, and Qi Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016.

Xi Zhou, Kai Yu, Tong Zhang, and Thomas S Huang. Image classification using super-vector coding of local image descriptors. In *European Conference on Computer Vision*, pages 141–154. Springer, 2010.

Güinter Zimmerman. Die hieroglyphen der Maya-handschriften, cram, 1956.

Gülcan Can

Contact Information

Phone: +41 27 721 77 85
E-mail: gulcan.can@idiap.ch
Website: http://www.idiap.ch/~gcan/

Education

PhD, Electrical Engineering, Swiss Federal Institute of Technology in Lausanne, Switzerland

- September 2013 - November 2017.
- Supervisors: Prof. Dr. Daniel Gatica-Perez and Dr. Jean Marc Odobez
- Project: Multimedia Analysis and Access for Documentation and Decipherment of Maya Epigraphy
- Field: Computer Vision, Machine Learning, Multimedia Analysis

M.Sc., Computer Engineering, Middle East Technical University, Ankara, Turkey

- October 2010 - September 2013.
- Thesis topic: Contextual Modeling of Remote Sensing Images with Conditional Random Fields
- Supervisor: Prof. Dr. Fatoş T. Yarman Vural
- Field: Computer Vision, Pattern Recognition, Remote Sensing
- Cumulative GPA: 3.57 over 4.00

B.S., Computer Engineering, Bilkent University, Ankara, Turkey

- September 2006 - June 2010.
- B.S. Project: Question answering system
- Course Project: Object recognition using SIFT keypoints and "bag-of-words" model
- Cumulative GPA: 3.21 over 4.00

High School, Ankara Science High School, September 2002 - June 2006.

Experience

Idiap Research Institute, Martigny, Switzerland

Research Assistant **September 2013 - December 2017**

Project: Multimedia Analysis and Access for Documentation and Decipherment of Maya Epigraphy

- Working on developing computer vision and machine learning algorithms in a multi-disciplinary cultural heritage project which engages several archeology and computer science partners. There are two main focus in my current research: crowdsourcing and representation learning for glyph shapes. Crowdsourcing part is about evaluating non-experts' perception of Maya glyphs. For glyph representation, knowledge-driven shape descriptors and data-driven representations (via sparse auto-encoders and convolutional neural networks) are evaluated.

Department of Computer Engineering, METU, Ankara, Turkey

Research Assistant **August 2010 - March 2013**

Project: Processing of Remote Sensing Imagery (Land Use/Land Cover Classification and Analysis)

- Performing active research in a computer vision/pattern recognition project which is run in close collaboration with several development teams in METU. The main work consists of developing state-of-the-art algorithms to be used in classification/detection of various types of regions/objects in satellite images. For that purpose, lots of hands-on experience is gained regarding various approaches in feature selection (texture features based on filter responses or gray-level co-occurrence matrix, histogram-based features such as HOG and LBP, keypoint features such as SIFT), clustering (k-means, mean-shift), segmentation (watershed, region-merging, mean-shift, graph-based), and classifier selection (SVM and kernel selection, decision tree, k-NN, Markov and conditional random fields).

Department of Computer Engineering, METU, Ankara, Turkey

Teaching Assistant **September 2011 - September 2013**

- Managing homeworks and labs for department courses such as data structures, introduction to programming (python), C programming, and providing guidance to senior design project groups.

Publications

- Gülcan Can, Jean-Marc Odobez, Daniel Gatica-Perez, *Visual Analysis of Maya Glyphs via Crowdsourcing and Deep Learning*, PhD Thesis, September 2017.
- Gülcan Can, Jean-Marc Odobez, Daniel Gatica-Perez, *How to tell ancient signs apart? Recognizing Maya glyphs with CNNs*, ACM Journal on Computing and Cultural Heritage (JOCCH), September 2017 (submitted).
- Gülcan Can, Jean-Marc Odobez, Daniel Gatica-Perez, *Shape Representations for Maya Codical Glyphs: Knowledge-driven or Deep?*, in Proc. 15th International Workshop on Content-Based Multimedia Indexing, June 2017.
- Daniel Gatica-Perez, Gülcan Can, Rui Hu, Stephane Marchand-Maillet, Jean-Marc Odobez, Carlos Pallan Gayol and Edgar Roman-Rangel, *MAAYA: Multimedia Methods to Support Maya Epigraphic Analysis*, Arqueologia computacional: Nuevos enfoques para el analisis y la difusion del patrimonio cultural, INAH-RedTDPC, 2017.
- Gülcan Can, Jean-Marc Odobez, Daniel Gatica-Perez, *Maya Codical Glyph Segmentation: A Crowdsourcing Approach*, IEEE Transactions on Multimedia, September 2017.
- Edgar Roman-Rangel, Gülcan Can, Stephan Marchand-Maillet, Rui Hu, Carlos Pallán Gayol, Guido Krempel, Jacob Spotak, Jean-Marc Odobez, Daniel Gatica-Perez, *Transferring Neural Representations for Low-dimensional Indexing of Maya Hieroglyphic Art*, in Proc. ECCV Workshop on Computer Vision for Art Analysis, Amsterdam, Oct. 2016.

- Gülcan Can, Jean-Marc Odobez, Daniel Gatica-Perez, *Evaluating shape representations for maya glyph classification*, ACM Journal on Computing and Cultural Heritage (JOCCH), Vol. 9, Issue 3, September 2016.
- Gülcan Can, Jean-Marc Odobez, Carlos Pallán Gayol, Daniel Gatica-Perez, *Ancient maya writings as high-dimensional data: a visualization approach*, in Proc. Digital Humanities, July 2016.
- Rui Hu, Gülcan Can, Carlos Pallán Gayol, Guido Krempel, Jacob Spotak, Gabrielle Vail, Stephan Marchand-Maillet, Jean-Marc Odobez, Daniel Gatica-Perez, *Multimedia Analysis and Access of Ancient Maya Epigraphy: Tools to support scholars on Maya hieroglyphics*, Signal Processing Magazine, IEEE, 32(4), pp.75-84, 2015.
- Gülcan Can, Jean-Marc Odobez, Daniel Gatica-Perez, *Is That a Jaguar? Segmenting Ancient Maya Glyphs via Crowdsourcing*, in Proc. ACM Int. Workshop on Crowdsourcing for Multimedia, 2014.
- Orhan Firat, Gülcan Can, Fatoş T. Yarman Vural, *Representation Learning for Contextual Object and Region Detection in Remote Sensing*, in Proc. Int. Conference on Pattern Recognition (ICPR), 2014.
- Gülcan Can, Fatoş T. Yarman Vural, *Contextual Modeling of Remote Sensing Images with Conditional Random Fields*, MSc Thesis, 2013.
- Gülcan Can, Orhan Firat, Fatoş T. Yarman Vural, *Conditional Random Fields for Land Use/Land Cover Classification and Complex Region Detection*, in Proc. 14th IAPR International Workshop on Structural and Syntactic Pattern Recognition (SSPR, jointly organized by ICPR), 2012.
- Ümit Ruşen Aktaş, Gülcan Can, Fatoş T. Yarman Vural, *Edge Aware Segmentation in Satellite Imagery: A Case Study of Shoreline Detection*, in Proc. 7th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS, jointly organized by ICPR), 2012.
- Ulya Bayram, Gülcan Can, Şebnem Düzgün, Neşe Yalabik, *Evaluation of Textural Features for Multispectral Images*, in Proc. SPIE Remote Sensing Conference, 2011.

Awards and Honors

- Best paper award at 3rd Computer Science Student Workshop (2012).
- Full-time scholarship by Bilkent University based on the ranking of ÖSS 2006 (Student Selection Exam), among 2 million candidates.
- Encouragement award in biology at 14th MEF Projects Competition in high school (2005)

Technical Background

Programming: C++, C, Java, MATLAB, Python, HTML, Javascript/Jquery
Frameworks/Libraries: Python numerical libraries (NumPy, SciPy, Scikit learn), Caffe, Matconvnet, Theano, Tensorflow, Keras
Technologies & Applications:

- Google App Engine: Coding back ends for an enterprise Java project.
- LaTeX, Microsoft Office, Open Office, and similar packages.

Operating Systems: Microsoft Windows, various Linux distributions.
Databases: MySQL.
Other: UML, XML etc.