Towards Query-by-eXpression Retrieval of Cuneiform Signs

Eugen Rusakov
Department of Computer Science
TU Dortmund University
44227 Dortmund, Germany
eugen.rusakov@tu-dortmund.de

Turna Somel
Hittitology Archive
Academy of Science and Literature Mainz
55131 Mainz, Germany
turna.somel@adwmainz.de

Gernot A. Fink
Department of Computer Science
TU Dortmund University
44227 Dortmund, Germany
gernot.fink@tu-dortmund.de

Gerfrid G.W. Müller
Hittitology Archive
Academy of Science and Literature Mainz
55131 Mainz, Germany
gerfrid.mueller@adwmainz.de

***Abstract*—Intermediate representations based on binary attributes are widely used for image retrieval tasks. Especially for the task of word spotting, attribute representations continually achieves state-of-the-art results. In contrast to Latin scripts, the cuneiform writing system is still a less well-known domain for the document analysis community. This writing system consists of wedge impressions, with the constellations and relative positions of wedges representing sign classes. While for Latin scripts characters are intuitive modeled as visual attributes, cuneiform signs do not reveal an evident approach to model visual attributes for cuneiform signs. However, in this work we introduce an attribute representation based on the so-called Gottstein-System. The idea is decompose signs according to wedge typology and enable a logical expression for sign classes, while sharing these expressions across visually similar cuneiform signs. We adapt this idea of describing wedge expressions in order to introduce to the Query-by-eXpression (QbX) retrieval scenario. Compared to queries based on sign IDs, our approach is capable of representing queries for an open-set retrieval scenario. Furthermore, we extend the Gottstein-System by expressions representing visual hints like wedge crossings or position relations. This way, promising results are achieved, as shown in our experiments.**

I. Introduction

Cuneiform is a writing system that was used from the end of the 4th millennium BCE until the 1st century in the Near East. It was written by pressing a stylus into moist clay tablets in order to create wedge-shaped impressions. A cuneiform sign is composed of one or more of such wedge impressions, and a sign may then be identified by the reader according to wedge direction, number, and positioning within the sign, similar to brush strokes in a Chinese character. A cuneiform sign may have syllabic or logographic values; as such, the writing system contains a greater number of signs than in an alphabet, with a maximum of ca. 900 signs. Due to the temporal and geographic range in which the writing system was used, its adaptation to at least 7 languages, as well as the personal preferences of scribe, both the inventories of regularly used signs and the form of the signs used, henceforth to be referred as sign variants, are heterogeneous. A sign variant may differ from other variants of said sign in all or some of the aforementioned characteristics, namely the direction, number, and positioning of the wedges. Since cuneiform studies is a philological field with a complex writing system, beginners heavily rely on sign lists, many of which contain a limited search representation based on a single 'main variant' of a sign class, listed according to the wedge shape and order. In order to simplify the retrieval process, Gottstein [1] proposes an alphanumeric representation system for cuneiform signs based on commonly occurring wedge types, represented by the letters **a**, **b**, **c** and **d**, and the count of each wedge type used in a sign.

In this work, we adapt the Gottstein-System and derive an attribute representation for the embedding of wedge types and their counts. Furthermore, we extend this system by including attributes based on visual hints like wedge type intersections which we named Last Elements. With our attribute based approach, we introduce a novel retrieval scenario, namely *Query-by-eXpression (QbX)*. Next to the well known Query-by-Example (QbE) scenario, our approach enables the retrieval based on user defined wedge expressions.

This work is organized as follows. In the next section related work is briefly reviewed. Section III-C introduces the Gottstein-System, and the resulting attribute representation. The experiments are discussed in Section IV. And finally, a conclusion is drawn.

II. Related Work

In this section a two sided related work in terms of cuneiform analysis and word embeddings is discussed. At first, we briefly review the related work for *cuneiform sign spotting*. Afterwards, we discus recently proposed *word embeddings* for word spotting.

A. Cuneiform Sign Spotting

Several approaches have been published for cuneiform sign recognition and spotting on 2D and 3D representations. In 2012, Mara [2] published an approach to extract 2D vector drawings from 3D cuneiform scans. Inspired by the *part-structured inkball models* proposed by Howe [3], Bogacz et al. [4] processed these spline graphs from [2] subsequently to match cuneiform signs using a similarity metric based on graph representations. Furthermore, in [5] the authors demonstrated how spline graphs can be applied as structural features for pattern matching. Massa et al. [6] followed a similar goal and described a method to extract graph representations from 2D cuneiform drawings. In [5] the authors presented an approach for segmentation-based cuneiform sign spotting based on part structured models. Besides the graph-based feature representations, Rothacker et al. [7] proposed a method using a *Bag-of-Features (BoF)* approach based on SIFT-Descriptors. Here, the segmentation-free cuneiform sign spotting is applied by integrating the *Hidden Markov Models (HMMs)* into a patch-based (sliding window) approach.

B. Embeddings for Word Spotting

A word embedding is a well known approach in the document analysis community, especially for the task of word spotting. In recent years different approaches have been published, proposing a certain embedding of word classes. In terms of handwriting, a user defined query can either be a visual image containing a handwritten word or a user typed string, which needs to be retrieved. Based on these query definitions, word spotting consists of two scenarios, namely *Query-by-Example (QbE)* and *Query-by-String (QbS)*.

The main challenge for the *QbE* scenario is to define a suitable measure of similarity for word image representations. For the *QbE* scenario there is no need to represent the word images in a human-interpretable fashion, hence, several query representations based on visual similarity have been proposed in the literature [8]–[10]. In contrast to the *QbE* scenario, a *QbS* scenario is defined by queries given as word string. The user is not required to search for a visual example of a word in the document collection to define a query. A drawback for the system is the requirement of a mapping from a textual to a visual (and vice versa) representation. Furthermore, for a model estimating word string representations, annotated word images are necessary for obtaining a *QbS* model.

A very popular approach called *Pyramidal Histogram of Characters (PHOC)* was presented in [8], which maps the visual domain into a textual representation. The authors proposed a word string embedding to project images of handwritten words into a common space based on a binary attribute representation using a pyramidal partitioning of a word string to include spatial information of character positions. In [11] Sudholt et al. proposed the *PHOCNet*, a CNN based on the *VGG* [12] architecture estimating attributes encoded in a PHOC representation by using a binary logistic regression in combination with a *binary cross entropy (BCE)* loss. Another evaluated embedding is the *Spatial Pyramid of Characters (SPOC)* [13]. In contrast to the PHOC, each split is built on a *Bag-of-Characters (BoC)* meaning a histogram of character counts. Wilkinson et al. [14] proposed an embedding based on the discrete cosine transformation applied to indicator matrix, namely *Discrete Cosine Transformation of Words (DCToW)*. The DCT is applied to each row of the matrix and the three largest coefficients from each row are than concatenated in order to form the final DCToW descriptor.

III. Attribute-based Cuneiform Sign Retrieval

In this section, we describe our approach of modeling the Gottstein-System as a binary attribute representation and including additional visual hints which we named Last Elements. Next to the representation, the database we are using is introduced.

A. Gottstein Representation

In [1] Gottstein proposed the idea of a unified representation for cuneiform signs, namely *Gottstein-Representation (GR)*. The author argues that sign lists and lexica used by researchers are not feasible to be used in a database, especially in terms of retrieval. Without existing knowledge of sign classes, searching through a cuneiform database becomes cumbersome. In order to simplify the search process, Gottstein [1] proposed

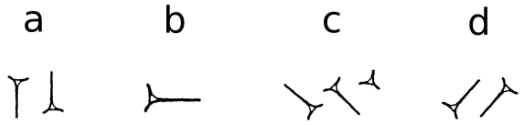

Figure 1. Hand drawn examples of wedge types for the alphanumeric representations **a**, **b**, **c** and **d**.

an alphanumeric representation system for cuneiform signs based on wedge types. The Gottstein-System follows a simple decomposition of wedge constellations into four different types denoted with **a**, **b**, **c** and **d**. For every letter the count of wedges is given. **a** and **b** represent vertical and horizontal wedges, respectively. The letter **c** represents a so-called *Winkelhaken* and an oblique wedge tending from the upper left to the lower right within a sign. And finally the letter **d** represents a wedge type which tends from the bottom left to the upper right direction. Figure 1 shows examples of wedge types for the corresponding alphanumeric representation. In this work, we

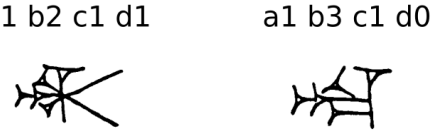

Figure 2. Hand drawn cuneiform signs with their corresponding Gottstein-Representations.

refer to the Gottstein-Representation of Hittite cuneiform signs provided by Michele Cammarosano from the University of Würzburg [15]. Figure 2 shows two cuneiform signs with their

corresponding Gottstein-Representations. The letters present wedge types, while the numbers represent the count of these types. As one can see, spatial information are missing. Based on this representation, one cannot derive the position for example of the wedge type **a** within the sign.

B. Last Element

Although the Gottstein-Representation provides a simple representation system to describe cuneiform signs, this simplification exhibits some shortcomings. Spatial information, which is an important feature in order to differentiate between different sign classes, is missing. For example if a sign consists of a horizontal wedge on the left side and a vertical wedge on the right side, the Gottstein-System is not capable of differentiating this type of wedge position. Furthermore, no relations between the wedges such as "intersecting each other" or "touching each other" are encoded in this system. Due to

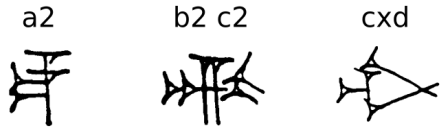

Figure 3. Hand drawn cuneiform signs and their corresponding last elements (visual hints).

these shortcomings, we extend the Gottstein-Representation by additional information based on visual hints, namely *Last Element (LE)*. When cuneiform signs are read from left to right, the wedge constellation on the far right represents the "last" constellation, hence the name last element. Although these last elements do not directly encode spatial information such as "a in the middle" or "b on the left side", they give some additional visual hints about the wedge constellation. Figure 3 shows three examples of additional visual hints. For example **a2** denotes two vertical wedges intertwining with each other. **b2 c2** means two horizontal wedges intertwine with each other and two Winkelhaken are on top of each other. On the right side of Figure 3 **cxd** represents an intersection between **c** and **d**. To avoid confusion, the last elements represent additional information encoded in seperate attributes as shown in the next Subsection (III-C).

C. From Gottstein to Attributes

In order to transfer the Gottstein-System into an attribute representation, we first derived the minimum and maximum counts of wedge types each. In the database we use in this work (see next Section III-D), the minimum count of wedge types represented by the letters **a**, **b** and **c** is 0, and the maximum count is 10. For the wedge type **d**, the minimum count is 0 and the maximum is 2. We model the count numbers of wedge types as single attributes, respectively. That means that our Gottstein attribute representation (GR) consists of a $10 + 10 + 10 + 2 = 32$ dimensional binary attributes vector ($\mathbf{a}_{max} + \mathbf{b}_{max} + \mathbf{c}_{max} + \mathbf{d}_{max}$). The last elements (LE) are encoded in the same way. The number of last elements we derived in addition to the Gottstein-Representation is 70 in total. Hence, the LE is modeled as a 70 dimensional binary attributes vector. Concatenating both, the encoded Gottstein-Representation (GR) and the last elements (LE) result in a $32 + 70 = 102$ dimensional attributes vector (GR + LE).

D. Database

In this work we use a set of annotated cuneiform tablets collected from a total of 130 photographs, provided by [16]. Figure 4 shows two examples of cuneiform tablets, which were inscribed by pressing a stylus while the clay was moist. Due to heavy damage through the millennia, tablets typically

Figure 4. Examples of cuneiform tablets. On the left side, a cuneiform tablet assembled from partially damaged fragments. On the right side, a image of a tablet taken from the side view.

have to be reassembled by philologists by joining fragments, as shown in the image on the left. The image on right-hand side demonstrates how the writing system typically exploits the whole surface of a clay tablet. For this collection, $44\,910$ samples of cuneiform signs were annotated by philologists [16], representing 291 different sign IDs. The sign images have a considerable variability in size as shown in Table I. Figure 5 shows examples of cuneiform signs. On the far left,

Table I
STATISTICS ABOUT IMAGE SIZES GIVEN IN PIXEL.

	max	*min*	*mean*	*std*
height	271	17	91	±30
widht	600	18	112	±47

the transliteration of a sign is given. Next to it an example of a hand drawing is shown. Each row shows four real examples cropped from the photographs. As one can see, the visual variability and quality in terms of wedge impression, image resolution and surface damage can be tremendous.

E. Model Architecture

Our model for cuneiform sign spotting is inspired by the successful *ResNet* architecture [17]. In the original work [17] 5 different versions were proposed. We adapted the *ResNet* architecture with 34 layers. The original *ResNet-34* uses strided convolutions for down sampling after the blocks 3, 7 and 13 in addition to the down sampling obtained from the pooling layer. As the cuneiform sign images are comparably small, we

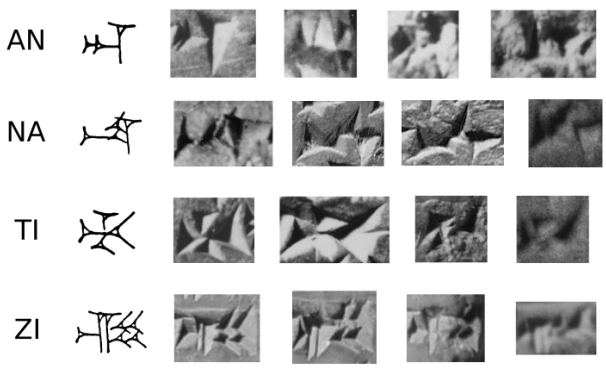

Figure 5. Examples of cuneiform signs with their corresponding transliteration and hand drawings.

decided to reduce the number of down sampling operations to preserve spatial information in the convolutional layers. Thus, we only use a down sampling in the first max pooling layer and another one after the 3 residual block. After the last convolution layer global average pooling is performed, which resizes the feature maps to a size of 1x1. The output after the global average pooling effectively results in a 512 dimensional feature vector. At the end, the feature vector is mapped to the attribute representation using a fully-connected layer. A major advantage of a global pooling after the last convolution layer is the capability to process an arbitrarily sized input image. As shown in Section III-D, the images differ in their sizes considerably. To avoid strong wedge distortions from rescaling within a cuneiform sign, we do not scale the images to a uniform image size. Due to different image sizes, we choose to delete all batch normalization layers [18]. We compute the gradient for mini-batches of varying sizes by obtaining the gradients for each image separately and then averaging them. Computing gradients in this way requires an effective batch-size of 1 and updating the weights after performing back-propagation n times where n is the batch-size. Thus, using a batch-normalization layer in that cast, a batch-size of 1 would be presented for determining the mean and variance. Obtaining feature maps with different sizes from different mini-batches would skew the computation of a global mean and variance value.

Table II
NUMBER OF SAMPLE FOR EACH CROSS-VALIDATION SPLIT.

	$Split_1$	$Split_2$	$Split_3$	$Split_4$	$Total$
#Samples	11 321	11 252	11 198	11 139	44 910

IV. EXPERIMENTS

For the experiments we use the database introduced in Section III-D, and a data partitioning as shown in Section IV-A. Section IV-B describes the evaluation protocol and Section IV-C explains the performance metrics. In Section IV-E we describe the training setup with all hyper-parameter used for training. Afterwards we discuss the retrieval results achieved by our approach in Section IV-F.

A. Data Partitions

We evaluate our method on the database described in Section III-D. As this data set does not have an official partitioning into training and test sets, we choose to perform a four-fold cross validation. Here, we sort the data set with respect to the classes and assign every fourth sample to one of the cross validation sets. Partitioning the data set in this way, we ensure that every set contains approximately the same amount of samples with equal sign IDs sets. Table II shows the number of samples from each cross validation split.

B. Evaluation Protocol

We evaluate our attribute prediction model for the cuneiform database in a segmentation-based cuneiform sign retrieval protocol similar to the word spotting standard protocol used for the *George Washington* data set [8]. The data set is partitioned into four cross-validation folds. Three cross-validation splits are used to train a single model, while retrieval is performed on the remaining split. For the Query-by-Example (QbE) scenario all test images, which occur at least twice in the test set, are considered as queries. Retrieval is performed by estimating the attribute representations for all test set sign images given the estimated attributes from a certain image query. The retrieval lists are obtained by sorting the distances of the list items in ascending order with respect to the query. For Query-by-eXpression (QbX) retrieval is performed equivalently to QbE except that a sign ID is only considered once as query. It is important to note that the relevance of an item within the retrieval list is determined by the relevance category, as we are performing the retrieval based on models either trained in a one-hot encoding fashion using sign IDs as targets or as a multi-class encoding using the Gottstein attributes representations as targets.

C. Evaluation Metrics

As performance metric, we use the interpolated *Average Precision (AP)* for each single query as in [13]:

$$AP = \frac{\sum_{i=1}^{n} P(i) \cdot R(i)}{t} \tag{1}$$

Where $P(i)$ is the precision if we cut off the retrieval list after the i-th element and $R(i)$ is the indicator function, which computes to 1 if the i-th position of the retrieval list is relevant with respect to the query and 0 otherwise. n denotes the length of the retrieval list. And finally, t is the total amount of relevant elements. Similar to segmentation-based word spotting tasks, the retrieval list contains all cuneiform sign images from the respective test set. Afterwards, the overall performance is obtained by computing the *mean Average Precision (mAP)* over all queries, where Q denotes the total amount of queries:

$$mAP_{query} = \frac{1}{Q} \sum_{q=1}^{Q} AP(q) \tag{2}$$

Table III
RESULTS FOR THE QBX AND QBE EXPERIMENTS IN MAP [%].

Sign IDs				QbX	QbE	
Loss	Similarity	Training Target	Relevance Category	$\text{mAP}_{\text{category}}$	$\text{mAP}_{\text{query}}$	$\text{mAP}_{\text{category}}$
CE	Cosine	Sign IDs	Sign IDs	**81.30**	**90.90**	**71.65**
			GR + LE	71.21	85.42	**72.56**
			GR	39.14	58.22	64.31
BCE	PRM	GR + LE	Sign IDs	62.93	77.85	53.20
			GR + LE	**79.45**	86.10	63.84
			GR	51.66	64.69	61.11
		GR	Sign IDs	31.66	77.85	53.20
			GR + LE	38.10	86.10	63.84
			GR	**86.23**	85.98	64.98
Cosine	Cosine	GR + LE	Sign IDs	56.47	79.61	53.71
			GR + LE	72.15	**88.30**	63.99
			GR	69.26	78.47	**65.43**
		GR	Sign IDs	30.19	49.58	28.10
			GR + LE	36.56	54.22	33.85
			GR	82.83	**87.14**	64.74

In addition to the typical performance metric *mAP* mostly used for word segmentation-based spotting tasks, we extend the evaluation by computing the *mAP* for every cuneiform sign ID separately and take the average over all sign IDs (denoted by C):

$$mAP_{category} = \frac{1}{C}\sum_{c=1}^{C} mAP(c) \tag{3}$$

As the data set is massively unbalanced, the query list is dominated by over-represented sign IDs. And as we are embedding the cuneiform signs into an attribute space mostly containing these over-represented IDs, the $\text{mAP}_{\text{category}}$ ensures a more balanced performance evaluation.

D. Similarity Measurement

For the segmentation-based scenario, retrieval is performed by computing a certain similarity between a given query and all test samples excluding the query itself. In this work we make use of two similarity measurements. The first one is the *cosine distance*, a well known and broadly used metric for retrieval tasks [11], [13], [14]. The cosine distance is defined as:

$$d_{cos}(\mathbf{q}, \mathbf{a}) = 1 - \frac{\mathbf{q} \cdot \mathbf{a}}{\|\mathbf{q}\|_2 \|\mathbf{a}\|_2} \tag{4}$$

Here, $\mathbf{q}$ denotes the query vector and $\mathbf{a}$ represents the estimated representation. This distance ranges between the values 0 and 1, were 0 represents highest similarity and 1 represents lowest similarity. The second similarity measure we use is the *Probabilistic Retrieval Model* presented in [19]. Here, the assumption is made that the binary attribute representation is a collection of d independent Bernoulli distributed variables, each having their own probability p of evaluating to 1.

The probabilistic retrieval model is defined as:

$$d_{prm}(\mathbf{q}, \mathbf{a}) = -\frac{1}{D}\sum_{i=1}^{D} q_i \log a_i + (1 - q_i) \log a_i \tag{5}$$

Similar to equation 4, $\mathbf{q}$ and $\mathbf{a}$ represent the query and the estimated attributes respectively, and D its dimensionality.

E. Training Setup

For the model training we use a *BCE* loss and the *Adaptive Momentum Estimation (Adam)* [20] optimizer. For the momentum the mean value β_1 is set to 0.9 and for the variance value β_2 is set to 0.999 while the variance flooring value is set to 10^{-8} as recommended in [20]. The batch size for all experiments is set to 10. We set the initial learning rate to 10^{-4} and divide the learning rate by 10 after $100\,000$ and another division by 10 after $150\,000$ iterations respectively, while running the training for $200\,000$ iterations in total. As parameter initialization, we use the strategy proposed in [21]. The weights are sampled from a zero-mean normal distribution with a variance of $\frac{2}{n_l}$, where n_l is the total number of trainable weights in layer l. Furthermore, we use some augmentation techniques to extend the samples in the database. At first we balance the training data by sampling the sign classes from a uniform distribution, followed by augmentation methods like perspective transformation, shearing, rotation, translation, scaling, lightness changing and noise generating techniques.

F. Results and Discussion

Table III lists the results for the QbX and QbE experiments on our cuneiform benchmark. The listed results are averaged over four cross-validations. The column with training targets shows the representation used during the training. For *Sign IDs*

we use a one-hot encoded vector where each sign ID is represented by a single vector element. GR and GR + LE denote the attribute representation for Gottstein and the Gottstein + Last Element representation respectively. The column which lists the relevance category shows results for retrieval performed with respect to a certain item relevance. In equivalence to the training target column, retrieved items are considered as relevant based on their mapping to sign IDs, GR, or GR + LE representations respectively. Every column showing the results contain three bold numbers which represent the best performance with respect to the relevance category. For the QbE scenario a one-hot encoded approach using the cross-entropy (CE) loss performs best for the relevance category sign IDs with 71.65% and GR + LE with 72.56% in terms of $\text{mAP}_{\text{category}}$, while the performance drops for the relevance category GR (64.31%). Retrieving with the relevance category sign IDs, the one-hot encoding outperforms our representations significantly. This is not surprising, as both GR and GR + LE have no information about sign IDs during training. Looking at the $\text{mAP}_{\text{query}}$ results, the one-hot encoding slightly outperforms the GR (87.14%) and GR + LE (88.30%) attribute representations with a retrieval performance of 90.90%. Here, the attribute presentations performs best using the cosine loss. For the QbX scenario promising results are achieved especially for the PRM model using the training target and relevance category GR (86.23%). Even though the results achieved by the model using one-hot encoding and the relevance category Sign IDs (81.30%) are not directly comparable, as two different relevance categories are used, we would still like to derive the conclusion that our representation based on wedge types expression fits well for a QbX scenario.

V. Conclusion

In this work, we introduced a novel retrieval scenario based on cuneiform wedge expressions namely, *Query-by-eXpression (QbX)*. We adapted the Gottstein-System and encoded it in a binary attributes representation. Furthermore, we extended this representation by visual hints, which we named *Last Elements*. We evaluated our approach on a database consisting of 130 photographs of cuneiform tablets. The experiments show that our proposed representation achieves similar and promising results compared to a one-hot encoded approach based on sign IDs. Besides the results, using a representation based on wedge expressions for cuneiform sign retrieval can be beneficial for experts searching in cuneiform databases for specific wedge constellations instead of sign IDs. Here, our proposed representation takes a first step toward cuneiform sign retrieval based on expressions. Nevertheless, further investigations need to be made in terms of a more powerful representation encoding more combinations of wedge constellations. One way to achieve such a representation could be the inclusion of spatial information with respect to the wedge positions or relations among them.

Acknowledgment

This work is supported by the German Research Foundation within the scope of the project 'Computer-unterstützte Keilschriftanalyse (CuKa)'

References

[1] N. Gottstein. (2012) Ein stringentes identifikations- und suchsystem für keilschriftzeichen. [Online]. Available: http://www.materiale-textkulturen.de/mtc_blog/2012_005_Gottstein.pdf

[2] H. Mara, "Multi-scale integral invariants for robust character extraction from irregular polygon mesh data," Ph.D. dissertation, 10 2012.

[3] N. R. Howe, "Part-structured inkball models for one-shot handwritten word spotting," in *Proc. of the ICDAR*, Aug 2013, pp. 582–586.

[4] B. Bogacz, M. Gertz, and H. Mara, "Cuneiform character similarity using graph representations," 2015.

[5] B. Bogacz, N. Howe, and H. Mara, "Segmentation free spotting of cuneiform using part structured models," in *Proc. of the ICFHR*, 2016, pp. 301 – 306.

[6] J. Massa, B. Bogacz, S. Krömker, and H. Mara, "Cuneiform detection in vectorized raster images," in *Computer Vision Winter Workshop (CVWW)*, 2016.

[7] L. Rothacker, D. Fisseler, G. G. W. Müller, F. Weichert, and G. A. Fink, "Retrieving cuneiform structures in a segmentation-free word spotting framework," in *Proc. of the Int. Workshop on Historical Document Imaging and Processing*, ser. HIP '15. New York, NY, USA: ACM, 2015, pp. 129–136.

[8] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *TPAMI*, vol. 36, no. 12, pp. 2552–2566, 2014.

[9] T. Rath and R. Manmatha, "Word spotting for historical documents," *Int. Journal on Document Analysis and Recognition*, vol. 9, no. 2–4, 2007.

[10] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, "Efficient segmentation-free keyword spotting in historical document collections," *Pattern Recognition*, vol. 48, no. 2, pp. 545 – 555, 2015.

[11] S. Sudholt and G. A. Fink, "PHOCNet: A deep convolutional neural network for word spotting in handwritten documents," in *Proc. of the ICFHR*, Shenzhen, China, 2016, pp. 277 – 282.

[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proc. of the Int. Conf. on Learning Representations*, 2015.

[13] S. Sudholt and G. Fink, "Evaluating word string embeddings and loss functions for CNN-based word spotting," in *Proc. of the ICDAR*, Kyoto, Japan, 2017, pp. 493–498.

[14] T. Wilkinson and A. Brun, "Semantic and verbatim word spotting using deep neural networks," in *Proc. of the ICFHR*, 2016, pp. 307 – 312.

[15] M. Cammarosano. (2012). [Online]. Available: https://www.hethport.uni-wuerzburg.de/HPM/hpm.php?p=3djoins

[16] Akademie der Wissenschaften und Literatur, Mainz and Julius-Maximilians-Universität, Würzburg. (2000) Hethitologie Portal Mainz. [Online]. Available: http://www.hethiter.net/

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. of the IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, p. 448–456.

[19] E. Rusakov, L. Rothacker, H. Mo, and G. A. Fink, "A Probabilistic Retrieval Model for Word Spotting Based on Direct Attribute Prediction," in *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, Niagara Falls, USA, 2018.

[20] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. of the Int. Conf. on Learning Representations*, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proc. of the Int. Conf. on Computer Vision*, 2015, pp. 1026–1034.