Zero-Shot Learning Based Approach For Medieval Word Recognition Using Deep-Learned Features

Conference Paper · October 2018

DOI: 10.1109/ICFHR-2018.2018.00067

6 authors, including:

Sukalpa Chanda
Østfold University College
52 PUBLICATIONS **549** CITATIONS
SEE PROFILE

Jochem Baas
University of Groningen
2 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Daniël Haitink
University of Groningen
2 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Dominique Stutzmann
French National Centre for Scientific Research
55 PUBLICATIONS **180** CITATIONS
SEE PROFILE

Zero-Shot Learning Based Approach For Medieval Word Recognition Using Deep-Learned Features

Sukalpa Chanda*, Jochem Baas*, Daniël Haitink*, Sébastien Hamel†, Dominique Stutzmann† and Lambert Schomaker*
*Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, Faculty of Science and Engineering,
University of Groningen, The Netherlands
Email:-s.chanda@rug.nl, l.r.b.schomaker@rug.nl
† Institut de recherche et d'histoire des textes, Paris, France
Email:-sebastien.hamel@irht.cnrs.fr, dominique.stutzmann@irht.cnrs.fr

***Abstract*—Historical manuscripts reflect our past. Recently digitization of large quantities of historical handwritten documents is taking place in every corner of the world, and are being archived. From those digital repositories, automatic text indexing and retrieval system fetch only those documents to an end user that they are interested in. A regular OCR technology is not capable of rendering this service to an end user in a reliable manner. Instead, a word recognition/spotting algorithm performs the task. Word recognition based systems require enough labelled data per class to train the system. Moreover, all word classes need to be taught beforehand. Though word spotting could evade this drawback of prior training, these systems often need to have additional overheads like a language model to deal with "out of lexicon" words. Zero-shot learning could be a possible alternative to counter such situation. A Zero-shot learning algorithm is capable of handling unseen classes, provided the algorithm has been fortified with rich discriminating features and reliable "attribute description" per class during training. Since deeply learned features have enough discriminating power, a deep learning framework has been used here for feature extraction purpose. To the best of our knowledge, this is probably the first work on "out of lexicon" medieval word recognition using a Zero-Shot Learning framework. We obtained very encouraging results(accuracy $\approx$57% for "out of lexicon" classes) while dealing with 166 training classes and 50 unseen test classes.**

I. Introduction

Historical manuscripts could provide information to reveal our past and make us aware of human societies that were prevalent during the historic period. To facilitate this process, historical documents are digitized and preserved in digital archives. Retrieving an information of interest in such archives usually leads to spotting/recognition of words based on which a document image can be indexed. But processing a historical document for OCR is more challenging compared to a contemporary handwritten document due to the following factors (a) different and challenging layouts and unwanted noise in the historical document image; (b) artefacts/torn historical document formation due to ageing with blurred, broken, faded text regions; (c) handwritten annotations in the text; (d) Contemporary handwritten characters in Latin (western scripts) are visually different from their medieval counterparts. These issues intensifies the problem of getting enough training data for every word-class. In some scripts the individual characters are fused to such an extent that they appear illegible. Under such adverse circumstances, the result of a character recognition based transcription is not reliable, hence such an approach cannot be used to index document images in a digital archive. Rather, indexing a document page with respect to a particular content can be achieved much reliably by either word spotting or by word recognition. In word spotting, a query (in form of an image or as a string of text characters) is searched for image regions containing similar words in the digital repository. Word recognition could retrieve a particular document image from a corpus only if a boundary estimator has drawn a bounding box in the region, and then only the recognizer can try to recognize the word inside that bounding box. Though only word classes from a priorly decided lexicon/dictionary could be used for the purpose of searching all document images. Word spotting through LSTMs is a bit more flexible in terms of dealing with "out of lexicon" words. Character-based LSTMs, can yield "out of lexicon" recognition, exploiting letter based compositionality. However, this can only be realized with a large and fully transcribed training set, at the line level. On top of that such LSTMs often need to consult a statistical "language model" for better accuracy. Many manuscripts are freshly digitized and the goal is to obtain access to the content as quickly as possible using any of those mentioned approaches.
Although Convolutional Neural Networks (CNN) have been successfully used for various tasks like image quality enhancement, image retrieval and writer identification, they cannot be readily deployed for "out of lexicon" medieval word recognition. However, from earlier experience their effectiveness in generating discriminating features cannot be ignored. The objective of the paper is to investigate the efficacy of a Convolutional Neural Network as a feature extractor while using Zero-Shot Learning algorithm as a classifier to deal with "out of lexicon" word classes.

II. Related Work

One of the early attempt to combine deep learning and zero-shot learning technique is due to [1]. In [1], a deep learning framework was used to generate features and consequently zero-shot learning was applied to classify between different

978-1-5386-5875-8/18/$31.00 ©2018 IEEE
DOI 10.1109/ICFHR-2018.2018.00067

animal, object and scenery images, three datasets namely- the Animals with Attributes dataset (AwA) [2], the aPascal/aYahoo objects dataset (aPY) [3], and the SUN scene attributes dataset (SUN)[4] were used in their experiments.

In [5] the authors investigated attribute label embedding methods for zero-shot and few-shot learning systems and [6] proposes a method that relies on human gaze as an auxiliary information generator. A benchmark and systematical evaluation of zero-shot learning w.r.t. three aspects, i.e. methods, datasets and evaluation protocol has been accomplished in [7].

In the recent past, some endeavours on word spotting are evident using CNN-based recognition framework. Sharma et al.[8] proposed a method where a pre-trained CNN is deployed to learn classes of word images. The output is then used to perform word spotting. The study claims that features extracted from an adapted-CNN can outperform hand-designed features on both spotting and recognition tasks for printed (English and Telugu) and handwritten document collections. However, their study does not consider historical document images. A very recent study by Sudholt et al.[9] proposed a novel CNN architecture for the purpose of word spotting, where both contemporary as well as historical document images were considered for experiments and encouraging results were obtained. Their proposed system can be customized as a "Query By Example"(QBE) or "Query By String"(QBS) based system. In [9] the network is trained with the help of a Pyramidal Histogram of Characters (PHOC) representation. The final layer of the network uses a sigmoid activation function that is applied to every element of the output vector. Another deep learning based approach for Arabic word recognition is due to [10]. A recent endeavour on deep learning based approach for indexing a vast medieval manuscript collection is evident in [11]. In the proposed method a deep recurrent neural network (RNN) and a statistical character language model was deployed to attain high accuracy in terms of word spotting and word indexing. From the brief discussion it is evident that though Zero-shot learning has been used extensively for animal, object and scenery image classification, it has been never used for word/text classification to the best of our knowledge. In this paper, we exploited techniques of Zero-Shot Learning for the purpose of an "out of lexicon" medieval word image classification task.

III. Motivation & Dataset Details

In a historical collection, it is quite obvious that occurrence frequency of different word classes will differ largely. Getting enough labelled instances from all possible word classes is itself a costly affair as it demands the intervention of a human expert. Moreover, sometimes an end user might be interested in searching for a word which is not at all present in the training set of the word recognizer. Hence it is practically impossible to train a system with all possible word classes. To counter this situation, a Zero-shot learning algorithm could be deployed which could classify "out of lexicon" words with a reasonably good accuracy. It is worth mentioning here that character-based LSTMs, can also perform "out of lexicon" recognition, at the cost of a large fully transcribed training set, at the line level and often with the aid of an appropriate "language model". But a Zero-shot learning algorithm does not need such extra onus to perform "out of lexicon" recognition.

The "original" word samples used in the experiments were procured from a large French administrative document collection of the medieval period. Those documents were produced due to French administrative activities during early 14th century and were digitized recently by the "French National Archive". Later, word samples were labelled in a crowd-source environment, where using a web interface a user (historian and or paleographer) could label a bounding box/word region in a document image, here an approach similar to [12] was followed. All crowd-sourced labelled word images were stored in a grey scale single channel pgm format. An off-the-shelf converter from [13] to convert those single channel images to 3 channel equivalent format with a size of 256×256 pixels was used. All 3 channels consist of the same grey scale values. The training set consists of 166 classes (with total 3338 images) whereas the test set consists of 50 classes (with total 756 images).

IV. Methodology

Though deep learning techniques have been extremely successful in dealing with various classification tasks like object classification, speaker identification and text recognition, it demands a huge amount of annotated training data per class in order to achieve high classification accuracy. In the context of medieval word image recognition, one must realize that this problem cannot be solved simply by adding contemporary handwritten data, as the shape of contemporary handwritten characters differs substantially from their medieval counterparts. Moreover, in general a deep learning algorithm could only classify a test sample to any of its training classes, thus failing completely when dealing with an "unseen" class sample. In a real world scenario, such pre-requisitions cannot always be met. For example, in the context of word recognition, it has been observed that some word classes are more frequent than others, and some word classes only have a handful of labelled instances, and in an extreme case the learning algorithm might need to classify a word image which does not have any labelled instances at all.

On the other hand, it cannot be ignored that deep learned features are in general very discriminative in nature. Hence, the proposed method makes an attempt to fuse the best of the two worlds. Here, the proposed method uses deep learned features along with a "Zero-shot Learning" framework to counter the problem of recognizing "out of lexicon" word class images, those word classes were never used in the training phase and were completely "unseen".

A. Zero-shot learning system

Zero-shot Learning is a relatively new branch of Machine Learning which is useful when there is no assurance of sufficient annotations for all possible class and objects to be classified. One must take into account that annotations/class label for some classes are easy to obtain in comparison to other classes in any image classification problem. This is due to scarcely available images for labelling which can only be made by a human expert in a particular domain. Zero-shot learning algorithms counter this situation by building a novel hybrid classifier with the amalgamation of a) existing classifiers and b) semantic, cross-concept mappings between class labels and visual appearance of an object class.

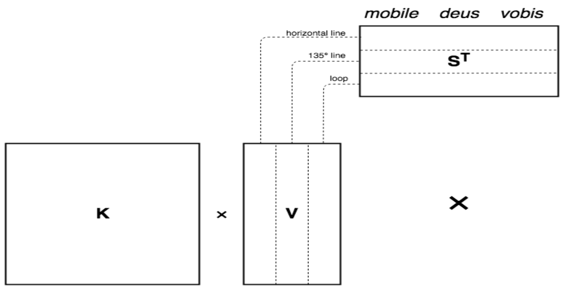

Fig. 1. The role of translation matrix V. Matrix V provides a mapping between the feature space in K and the attribute space in S.

The proposed zero-shot learning algorithm is based on *Embarrassingly Simple Zero-Shot Learning* as described in [1]. The system has been developed in python and is available in a git repository[1]. We will use an adapted notation scheme here in comparison to [1].

During the training stage, the system first creates a kernel $K \in R^{m \times m}$ out of the instance matrix $X \in R^{m \times d}$, where m is the number of instances and d is the dimensionality of the data. This kernel can be computed linearly (as seen in figure 2), or as a Gaussian kernel which depends on the hyper-parameter σ. Equation 1 shows the computation for the Gaussian kernel K.

$$K(X_i, X_j) = exp\left(-\frac{\|X_i - X_j\|^2}{\sigma^2}\right) \tag{1}$$

Together with the signature matrix $S \in [0,1]^{z \times a}$, where z is the number of word classes, and a is the number of signature attributes (as explained in IV-C), the system computes a matrix V which maps between the feature space (represented by K), and the attribute space (represented by S), according to equation 2, where $Y \in \{0,1\}^{m \times z}$ represents the ground truth labels of each instance belonging to any of the z word classes.

$$V = (K^\top K + \gamma I)^{-1} K Y S (S^\top S + \lambda I)^{-1} \tag{2}$$

In equation 2, λ and γ are hyper-parameters. Together with hyper-parameter σ, these are optimised during learning. These parameters are denoted as:

- The value of σ represents the standard deviation of the Gaussian kernel computation.
- The value of λ makes the instances on the attribute space KV more invariant. This improves the total accuracy [1].
- The value of γ balances the values of signature attribute matrix S. If the attribute values are unbalanced, the system may not perform optimally [1].

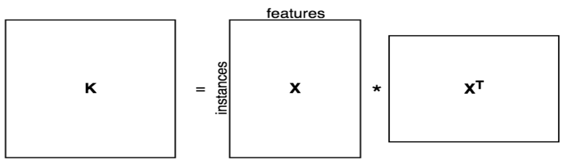

Fig. 2. Linear computation of matrix K from matrix X [1].

During training, the optimal values for hyper-parameters λ and γ and σ(only while using Gaussian Kernel) were determined using n-fold cross validation technique. Those optimal values for the hyper-parameters were consequently used during the testing phase. During the inference stage, a new set of classes (where z' denotes total number of test classes) were introduced, with their attribute signature in matrix S', and with their instances in matrix X'. Linearly, the kernel K' is then computed as: $K' = X'X^\top$. Note the relation between the training instances (in matrix X) and the testing instances (in matrix X'). Similarly, the Gaussian kernel variant of kernel K' can be computed using the optimal σ value found during training. The resulting classification is calculated per instance k in K' by:

$$\operatorname*{argmax}_i kVS_i'^\top \tag{3}$$

In equation 3, i represents the class out of z' that the instance k is classified as.

B. Deep Learning as a Feature Extraction Tool

The AlexNet CNN architecture [14] is made up of two parts: (a) the first few layers of the network performs the convolution with different filters and acts as a feature generator; (b) the features generated in the first few layers are propagated through the deeper layers - which are known as fully connected layers. Here, multiple fully connected layers are stacked together to form the Multi-Layer Perceptron architecture. The AlexNet architecture was customized to adapt to the word recognition problem by setting the crop size to 256×256 and by not considering the flipping option for this modified CNN setup. Optimal network weights were obtained using a Stochastic Gradient Descent (SGD) technique with an initial learning rate of 0.001 along with an inverse decay function to lower the learning rate proportionally with respect to the increase in the number of iterations. The max. number of iterations while training, was set to 100,000.
For the training stage of the network, the training instances are divided into a training and a validation part, with the validation part containing 20% of the total training instances. Before training, all word images are re-sized to 256 by 256 pixels using an off the shelf converter from [13]. To see what the effects were of the number of training iterations, we ran

[1]https://github.com/Zero-Shot/

tests with the network trained for 10,000, 50,000 and 100,000 iterations. The feature vectors were tapped from the "FC6" layer of the AlexNet architecture for all experiments.

C. *Attribute signatures for word recognition*

In the context of Zero-shot learning "class/attribute signatures", represent semantic information about classes involved in the training and testing of the system. A "class/attribute signatures" represents some unique visual/semantic characteristics of the associated class which clearly makes a distinct mark of the difference with other classes involved in the classification process. For example, to recognize/classify/differentiate between a tiger and other wild carnivorous animals, the presence of black stripes on the body of the tiger could be used as one of the most obvious "class/attribute signature". The value for this "class/attribute signature" could be set to 1 for the "Tiger class" and 0 for other classes. Since images of Text/Words are devoid of such glaring visual clues, we need to rely on the presence of primitive shape structures in the shape of the word, to procure different "class/attribute signature" values. The primitive shape structures that we have used to define a "class/attribute signature" for a particular word are different types of handwriting strokes. They are as follows:(a) Ascending line - If there is a line that extends above the height of "x-height", we consider it an ascending line; (b) Descending line - If there is a line that extends below the bottom of "x-height", we consider it a descending line; (c) Small left semi-circle; (d) Small right semi-circle; (e) Large left semi-circle; (f) Large right semi-circle; (g) Full circle - If there is a full circle, then there are also two semi-circles. The semi-circle attributes carry the information about the size, so the full circle attribute does not need a small or large variant; (h) Vertical Line; (i) Diagonal line (45°), going from bottom left to top right; (j) Diagonal line (135°), going from bottom right to top left; (k) Horizontal line.

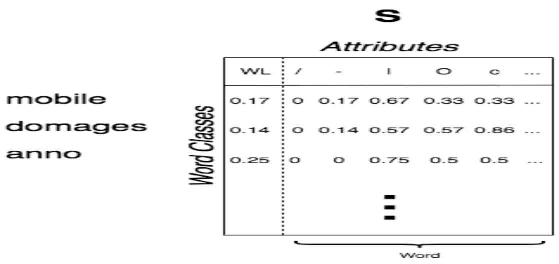

Fig. 4. Example of matrix S

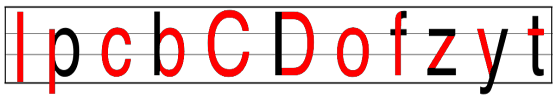

Fig. 5. The eleven types of basic shape attributes, highlighted in red. From left to right: ascender, descender, left small semi-circle, right small semi-circle, left large semi-circle, right large semi-circle, circle, vertical line, diagonal line (45°), diagonal line (135°), horizontal line.

The "class/attribute signatures" of a word class are computed by considering different shape attributes of its characters. Multiple variants of "class/attribute signatures" matrix were derived using different type of combinations of those 11 primitive shape attributes. Given a word class, the scores for each of those primitive shapes were computed using a simple software module. The variants of "class/attribute signatures" matrix are as follows:(a) S - this is the most basic "class/attribute signatures" matrix consisting of just the occurrence count of each primitive shapes and one additional column with information about the total number of characters present in the word(this word length information attribute is common to the other variants of the "class/attribute signatures" matrix). Finally, the scores are normalized by the total number of characters in the word; (b) *4S* - This matrix is computed by first dividing the length of the words into 4 parts and then computing scores for 11 primitive shape attributes within each of those 4 parts. When dealing with word lengths (total number of characters in the word) that are not divisible by 4, we decide the splitting points of the word based on the lowest integer found after dividing the word length by 4. Hence in case of a word with a word-length that is not exactly divisible by 4, the last parts will be bigger than the first. At most, the difference will be of one character. In the case of a word with less than four characters, the first parts of the 4S matrix will remain "empty", as it will be filled with zeros; (c)*S-Alphabet* - This matrix consists of the occurrence count of each primitive shapes, normalized by the total number of characters in the word, additionally it considers the occurrence count for each individual alphabet; (d)*4S-Alphabet*- This matrix is computed by considering 11 primitive shapes and the occurrence count for each individual character within each of the 4 word parts. Details on each matrix type and properties are in Table I. Word class "class/attribute signatures" computed in *S* and *S-Alphabet* do not take into account the ordering of the appearance of primitive shape attributes in the word. To counter this drawback, in *4S* and *4S-Alphabet* the word is first divided into four parts, and then within each part, the presence of each shape attribute is computed. The final score for each signature attribute is normalized by the total number of characters in each part. These word class attribute signatures do take into account character order in a general way, but at the cost of a longer vector dimensions.

V. Experimental Results

Experiments were conducted under several different experimental setup, which involves (a) different versions of the signature attribute matrix S; (b) different number of test classes; (c) the use of a Gaussian kernel or a linear kernel for kernel K. In order to analyze the effect of the training iterations on system performance, experiments were conducted on features tapped from the FC6 layer of the AlexNet architecture after 10,000, 50,000 and 100,000 training iterations of the AlexNet.

As it can be observed from figure 6 the accuracy of the system depends mostly on the choice of the type of signature-attributes in matrix S. The *4S-Alphabet* type of signature

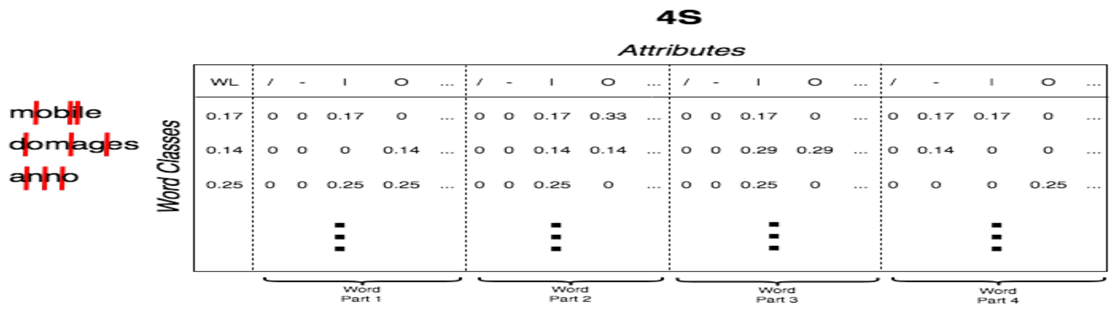
4S
Attributes
WL / - I O ... / - I O ... / - I O ... / - I O ...
mobile 0.17 0 0 0.17 0 ... 0 0 0.17 0.33 ... 0 0 0.17 0 ... 0 0.17 0.17 0 ...
domages 0.14 0 0 0 0.14 ... 0 0 0.14 0.14 ... 0 0 0.29 0.29 ... 0 0.14 0 0 ...
anno 0.25 0 0 0.25 0.25 ... 0 0 0.25 0 ... 0 0 0.25 0 ... 0 0 0 0.25 ...
Word Classes
Word Part 1
Word Part 2
Word Part 3
Word Part 4

Fig. 3. Example of matrix S in four parts. The red lines show the division of the words in the four parts.

TABLE I
NUMBER OF ATTRIBUTES OF THE FOUR VERSIONS OF SIGNATURE ATTRIBUTE MATRIX AND THEIR PROPERTIES. S

Type of signature/attribute matrix (attribute dimension)	**Properties of the matrix**
S (12)	One column denoting the number of characters present in the word and scores for 11 primitive shape attributes in a word.
S-Alphabet (38)	Same as above, additionally taking into account the presence of each alphabet and dividing the occurrence count of each alphabet by the total number of characters in the word.
4S (45)	The word image is divided into 4 parts and within each part scores for 11 primitive shape attributes in a word was computed. Hence 11×4 makes 44 attributes plus another column denoting the number of characters present in the word.
4S-Alphabet(149)	The word image is divided into 4 parts and within each part scores for 11 primitive shape attributes in a word along with the count of 26 alphabets were computed. Hence 37×4 makes 148, plus one column denoting the number of characters present in the word.

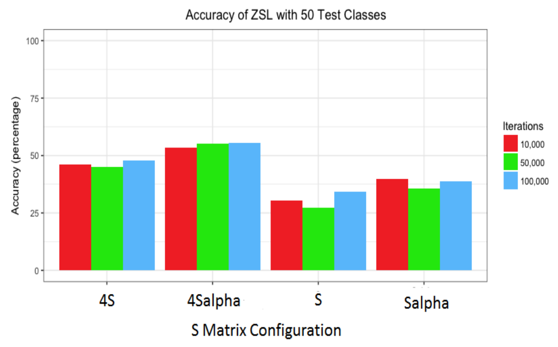

Fig. 6. Accuracies (%) using Gaussian kernel K for all four types of S for 50 unseen test classes over a different number of training iterations.

matrix shows the highest accuracy. Interestingly, the number of training iterations of the AlexNet does not seem to have a large impact on the accuracy of the system. This could be due to the fact that the number of instances is quite low and hence the network does not improve much as with higher iterations it is basically seeing the same image multiple times.

Also it is worth mentioning that the accuracies obtained by the use of the Gaussian kernel (as seen in figure 6) are a bit higher compared to the accuracies obtained by the use of the linear kernel. The highest accuracy obtained with a Gaussian Kernel is 56.87%, whereas with a linear Kernel it is $\approx$ 51%.

A. *Effect of Data Augmentation*

Since the training dataset used in this experiment was rather small, it is worth to investigate the system performance where the system has been trained with more data. To meet this demand, an off-line data augmentation technique based on "elastic morphing" were used alike the method in [15]. Using that technique we obtained 34,739 training instances in total (which includes both augmented and real image samples), however, no test instance was processed with the off-line data augmentation technique. The reason for not augmenting test data is that the results on synthetic data might not be that reliable to gauge the real system performance. Hence, for this experiment the number of test instances was the same as before, that is 756 (from 50 classes). Under this experimental setup with a linear kernel the accuracy obtained was 52.60% and while using a Gaussian Kernel it became 55.95% (using features from the network after 300k iterations), which is bit lower than the accuracy obtained while using only "original images".

VI. DISCUSSION & ANALYSIS

A. *Comparison with previous research*

Since this is the very first work on word recognition using a Zero-Shot Learning framework, we cannot compare our work with any published research directly. However, it can be noted from Table II that the proposed system gets comparable results to the results found in [1]. Here in Table II, (Tr) denotes training and (Te) denotes testing. The experiments performed by [1] used regular image data sets, with images of animals, objects and scenery. With a test dataset which contains four to five times as many unseen classes(50), the proposed system achieves an accuracy of 56.87%, when the AlexNet was trained for 100,000 iterations, and the type *4S-Alphabet* for matrix S was chosen. With a test set which contains 10 unseen classes, our system achieves an accuracy of 77.51%, which is (significantly) higher than the method in [1]. In Table II, the first three columns from the left denotes accuracies obtained by [1] on their dataset, and the last two columns from the right state accuracies obtained from the proposed method in the context of medieval word recognition.

TABLE II
OUT OF LEXICON PERFORMANCE USING ZSL ON REGULAR IMAGE DATASETS(ANIMALS, OBJECTS) AND OUR HISTORICAL HANDWRITTEN WORD IMAGE DATASET.

	AwA	aPY	SUN	*Medi-50*	*Medi-10*
Tr. Classes	40	20	707	*166*	*206*
Te. Classes	**10**	**12**	**10**	***50***	***10***
Accuracy	49.30%	27.27%	65.75%	*56.87%*	*77.51%*

B. Error Analysis

It was observed initially that a considerable number of misclassifications occurred between very similarly shaped word pairs, for e.g. "alios" and "alias", or "bonne" and "bono". For a visual illustration please refer to Fig.7. From this incident, it can be realized that the learnt feature space to attribute space mapping is quite robust, otherwise misclassification between such very similarly shaped word pairs would not have occurred so frequently. Apprehending low training samples from the training set as the root cause of this problem, training was done with augmented samples as well, however, a similar misclassification trend was still observed even after training was accomplished using the augmented image samples. One must consider the fact that medieval handwriting appears different than contemporary handwriting, hence computing the signature/attribute values on those medieval word images seems to be a bit error prone. Computing the signature/attribute values on contemporary word images would have generated more reliable signature/attribute values and hence would have given much higher accuracy on contemporary handwritten images while dealing with "unseen word classes".

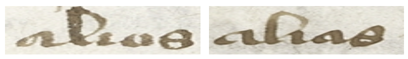

Fig. 7. Example of two near similar shaped words those got misclassified frequently.(left)Alios,(right) Alias.

VII. CONCLUSION & FUTURE WORKS

This work proposes a method to recognize "unseen word class" instances improvising a Zero-Shot Learning algorithm. Deep learned features tapped from the FC6 layer of AlexNet was used as features along with different primitive shape structures (like vertical line, horizontal line, half right circle, half left circle etc) to be used as elements of the attribute matrix of the Zero-Shot Learning algorithm. Encouraging results of $\approx 57\%$ were obtained while dealing with 50 unseen classes from medieval word images. Future research might involve features from more advance CNN architectures like ResNet and GoogleNet and to further customize the Zero-shot learning algorithm to consider One-shot learning problems as well.

VIII. ACKNOWLEDGEMENTS

This research work has been funded by NWO/Creative Industries grant (Project No.335-54-001) under the aegis of the "HIMANIS - Historical MANusscript Indexering for user-controlles Search" an EU project (jointly funded by the Dutch, French, Spanish research authorities in the framework of the enquoteJoint Programming Initiative on Cultural Heritage and Global Change)

REFERENCES

[1] Bernardino Romera-Paredes and Philip Torr, "An embarrassingly simple approach to zero-shot learning," in *Proceedings of the 32nd International Conference on Machine Learning*, Francis Bach and David Blei, Eds., Lille, France, 2015, vol. 37 of *Proceedings of Machine Learning Research*, pp. 2152–2161.

[2] CH. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR 2009*, Piscataway, NJ, USA, June 2009, Max-Planck-Gesellschaft, pp. 951–958, IEEE Service Center.

[3] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth, *Describing objects by their attributes*, pp. 1778–1785, 2009.

[4] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 2751–2758.

[5] Zeynep Akata, Florent Perronnin, Zaïd Harchaoui, and Cordelia Schmid, "Label-embedding for image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1425–1438, 2016.

[6] Nour Karessli, Zeynep Akata, Bernt Schiele, and Andreas Bulling, "Gaze embeddings for zero-shot image classification," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 6412–6421.

[7] Yongqin Xian, Bernt Schiele, and Zeynep Akata, "Zero-shot learning - the good, the bad and the ugly," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 3077–3086.

[8] Arjun Sharma and K. Pramod Sankar, "Adapting off-the-shelf cnns for word spotting & recognition," in *13th International Conference on Document Analysis and Recognition, ICDAR 2015, Nancy, France, August 23-26, 2015*, 2015, pp. 986–990.

[9] Sebastian Sudholt and Gernot A. Fink, "Phocnet: A deep convolutional neural network for word spotting in handwritten documents," in *15th International Conference on Frontiers in Handwriting Recognition, ICFHR 2016, Shenzhen, China, October 23-26, 2016*, 2016, pp. 277–282.

[10] Alex Graves and Juergen Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., pp. 545–552. 2009.

[11] Théodore Bluche, Sebastien Hamel, Christopher Kermorvant, Joan Puigcerver, Dominique Stutzmann, Alejandro Héctor Toselli, and Enrique Vidal, "Preparatory KWS experiments for large-scale indexing of a vast medieval manuscript collection in the HIMANIS project," in *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017*, 2017, pp. 311–316.

[12] Tijn van der Zant, Lambert Schomaker, and Koen Haak, "Handwritten-word spotting using biologically inspired features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1945–1957, 2008.

[13] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22Nd ACM International Conference on Multimedia*, 2014, MM '14, pp. 675–678.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105. 2012.

[15] Marius Bulacu, Axel Brink, Tijn van der Zant, and Lambert Schomaker, "Recognition of handwritten numerical fields in a large single-writer historical collection," in *10th International Conference on Document Analysis and Recognition,ICDAR 2009, Barcelona, Spain, 26-29 July 2009*, 2009, pp. 808–812.