Experiment study on utilizing convolutional neural networks to recognize historical Arabic handwritten text

Reem Alaasam*, Berat Kurar*, Majeed Kassis* and Jihad El-Sana*
Department of Computer Science
Ben-Gurion University of the Negev
Beer-Sheva, Israel
rym, berat@post.bgu.ac.il, majeek@cs.bgu.ac.il, el-sana@cs.bgu.ac.il

***Abstract*—Deep learning is a form of hierarchical learning, it consists of multiple layers of representations that gradually transform data into high level concepts. Deep learning has been providing the state of the art results for various computer vision problems. However, a typical deep leaning algorithm needs a large amount of data to train a deep model and guarantee the models ability to generalize. It is not easy to generate large labeled datasets and it is one of the main barriers to apply deep learning for many problems. Data augmentation schemes were introduced to overcome this limitation, by extending small available labeled datasets. In this work we experiment with extending a small labeled dataset of Arabic continuous subwords by an orders of magnitude. The labeled dataset, which consist of handwritten Arabic subwords is used to synthesize a large collection of labeled dataset. The synthesized subwords are based on one or multiple writing styles from the original labeled dataset. We also experiment with generating various printed forms of subwords. We include only Naskh font, as most of the Arabic historical manuscripts were written in this type of font. We train several convolutional neural networks using handwritten, printed and synthesized datasets and obtain encouraging results.**

I. Keywords

Database, Arabic, handwritten, text recognition.

II. Introduction

Historical documents have been providing an interesting peek into the history of the past societies. Over the last two decades many archives have been scanned and huge collections of document images have been generated. The availability of these collections is attracting the interest of researchers and many algorithms were developed to process historical document images. A system that is able to read historical document images accurately and provide users with a textual query is crucial. Using such a system enables scholars to study the historical documents extensively.

Historical handwritten document images are different from the modern document images by their loosely layout format. They contain overlapping components in a line. Words have writer dependent varying shapes. They include holes, spots, ornamentation or seals that degrade overall quality significantly [1]. Optical Character Recognition (OCR) systems which make recognition character by character, are designed for modern printed documents and do not perform well in processing historical handwritten document images.

An alternative solution is the recently proposed method, word spotting [2]. The idea is to segment a page into words instead of characters. Word spotting on historical handwritten document images is a classification task on the words included in the document. Word spotting systems usually employ hand crafted features to represent the image data [3]. However learning features on large images is computationally expensive and increases the number of parameters which in turn makes the overall process slower.

Recently the models based on deep convolutional networks have dominated the visual recognition tasks. Deep models have been successfully applied to problems such as text recognition [4], image recognition [5] and character recognition [6]. Convolutional Neural Networks (CNN) consist of convolutional layers and pooling layers followed by fully connected layers. This architecture uses the 2D structure of an image as direct input without the need to hand crafted features for representing image data. The filters in convolutional layers are automatically learned by the backpropagation process in the network. However CNNs require large datasets for training due to their deep structures in order to prevent overfitting.

In this paper we explore utilizing CNN to classify Arabic subwords and study two ideas to overcome large dataset limitation. Firstly, we extend a small labeled dataset of handwritten Arabic subwords, by automatic synthesis of these subwords [7]. The synthesized subwords are based on one or more writing styles from the original labeled dataset. Secondly, we extend this labeled dataset by generating various printed forms of subwords using the Naskh font as most of the Arabic historical documents were handwritten using variation of this font. We trained CNN using the extended datasets and comparatively show their contributions.

In the rest of the paper we present related work about Arabic text recognition and word spotting in section 3, the dataset extension methods and the datasets' statistics in section 4, results of our experiments in section 5 and lastly in section 6 we draw conclusion and propose a future work.

III. Related Work

The recognition of Arabic text has attracted the interest of researchers over the last two decades and many quality papers has been published. Early works assume that Arabic characters can be isolated individually. First group of approaches extract outer contour or the skeleton of the characters to obtain Fourier descriptors [8], [9], [10] or rely on the Fourier coefficients of the handwritten dynamic representation [11]. Another approach uses class conditional density functions of Arabic characters to make Bayes classification [12]. Segmentation based methods separate words into characters based on their geometrical and topological properties. [13], [14]. Other approaches segment the words into characters by vertical projection and histogram techniques [15], [16]. Segmentation was also made based on HMM models [14] or morphological rules which are constructed at the feature extraction [17]. Recently the leading methods for recognizing recursive script became segmentation free. Maddouri and Amiri [18] introduced global features specific to Arabic and rate the recognition system by propagating these features into a transparent neural network. Saabni [19] presented a multi-level recognizer for online Arabic handwriting in a holistic fashion, thus avoided segmenting words into individual letters.

Word spotting was initially proposed in [20] and it aims to detect a word in the printed or handwritten document image. Word spotting utilizes word matching algorithm to measure the distance between representations of the words. There are two types of word matching: pixel-based and feature-based. The former measures the similarity of two images using Euclidean Distance Map, XOR difference, Scott and Longuet-Higgins distance, Hausdorff distance, sum of square differences [21], [22], [23] or etc. The latter measures the similarity of two images using the features extracted form the images [24], [25], [26], [27].

Recent document processing algorithms extract interest points directly from gray scale images [28] and utilize these points for various applications, such as word spotting [29], [30] and writer identification [31]. Most of these algorithms impose a grid or define patches to control the distribution of feature points [32], [29]. Defining the size of this grid and the number of sample points is done in an ad-hoc manner. These algorithms demonstrate improvement over binary-prerequisite-based algorithms for gray scale images. Other works are based on bag-of-visual-words model, such as [28], [33], [34], [35]. The performance of these algorithms deteriorates as the degradation level increases [36]. The idea of using these points to compare similarity among components is based on the hidden assumption that these points faithfully represent the processed text components.

IV. Experimental Study

In this section we describe the two methods that are used to extend a labeled dataset, the statistics of the three datasets that are used in our experiments and the architecture of the convolutional network we trained to classify these labeled datasets.

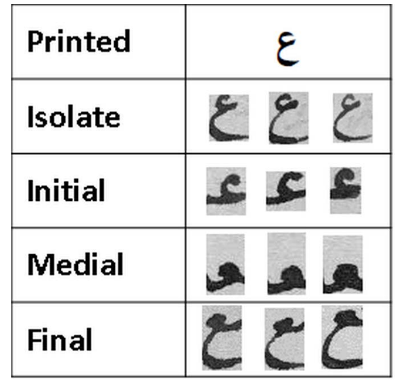

Fig. 1. An example for an LCM entry, and its output in various positions

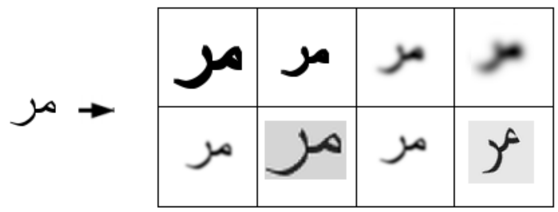

Fig. 2. Some examples of degraded printed text in the Naskh font.

A. Automatic synthesis of historical Arabic subwords

Automatic synthesis of historical handwritten Arabic subwords is a novel framework that is proposed recently by [7]. It first builds a Letter Connectivity Map (LCM) (Figure 1) that includes multiple instances of each letter's various shapes, since an Arabic letter's shape varies by its position in the word. The LCM is generated from several historical pages that were annotated previously. This LCM is then used to guide the automatic synthesis of any Arabic continuous word, from its ASCII representation.

B. Generating printed forms of Arabic subwords

The image of a printed text is perfect and is not suitable to be trained with for learning historical handwritten text images. Therefore we introduce various kinds of degradations into the printed text images by rotation, blurring, filtering and zooming. The Naskh font is used to type the printed text because the Arabic historical documents were commonly written in this font. Figure 2 demonstrates some examples.

C. Datasets

We created three datasets (Table I), the historical handwritten text image dataset (HD), the synthesized text image dataset (SD) and the printed text image dataset (PD). The handwritten text image dataset is extracted from an original historical document (Figure 4) in [37]. The synthesized text image dataset is automatically synthesized by the method proposed in [7]. The printed text image dataset is generated by various forms of subwords in the Naskh font.

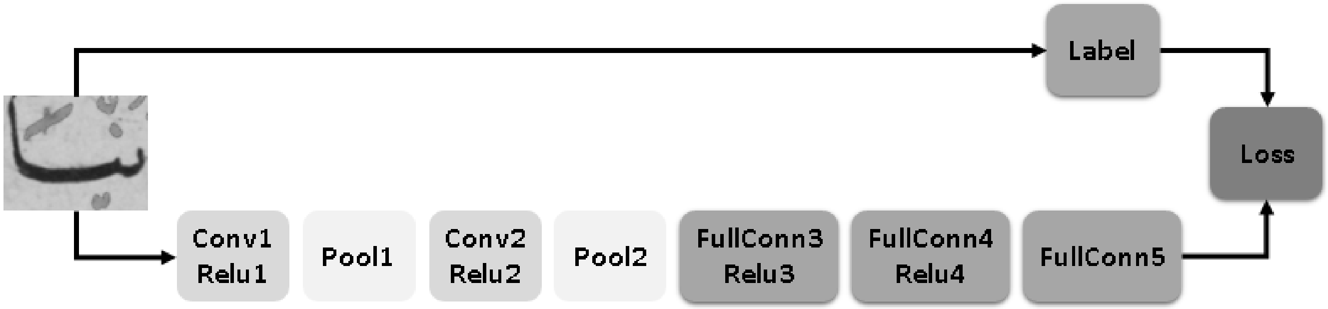

Fig. 3. The architecture of the CNN used in the experiments

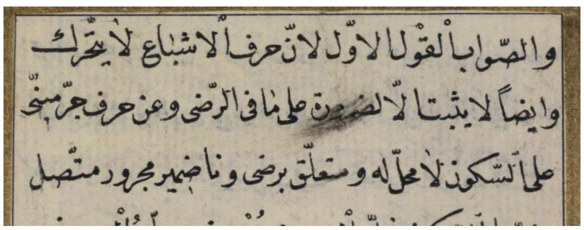

Fig. 4. A page part from the historical document that is used in the experiment.

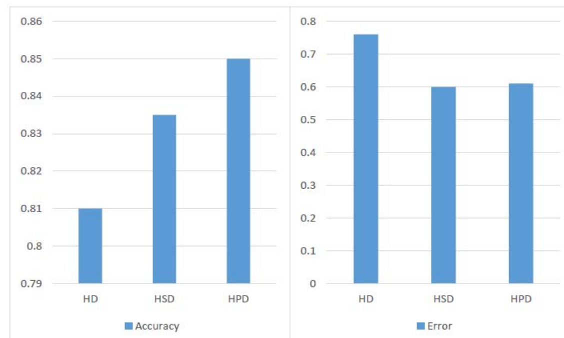

Fig. 5. Accuracy and error values of the 3 experiments. The horizontal axis labels show the training sets.

In all our experiments datasets are separated into training dataset S_{train} and testing dataset S_{test}. S_{train} and S_{test} are independent where

$$\emptyset = S_{train} \cap S_{test}$$

TABLE I
DETAILS OF THE DATASETS

	Train set	Test set	# of classes
Handwritten	51853	22218	68
Synthesized and Handwritten	53784	22218	68
Printed and Handwritten	74071	22218	68

D. The architecture

For the experiments in this paper we use a deep learning architecture shown in Figure 3. It is a 5 layer CNN with two convolutional layers, two fully connected layers and one final fully connected layer. We scaled the input images to 100×100 pixels. The CNN model was implemented using the Keras library [38] on Theano [39] backend. It was trained using a stochastic optimization method ADAM [40] with a learning rate of 0.01, and with exponential decay rates of 0.9 and 0.999. Using a five core CPU, training took 12 hours on the average with a batch size of 32.

V. Results

We performed three experiments. First experiment is performed on the handwritten text image dataset (HD). Second experiment is performed on the combination of the handwritten text image dataset and the printed text image dataset (HPD). Third experiment is performed on the combination of the handwritten text image dataset and the synthesized text image dataset (HSD). For a fair comparison; the same CNN architecture and the same optimization algorithm are used with each of the datasets. Results are compared by the evaluation measures error and accuracy on HD test set. The purpose of testing on HD test set is to expose the practical performance of the experiments.

First experiment is trained on HD train set and tested on HD test set. This experiment is aimed to evaluate the performance of CNN as shown in Figure 5 we got 81% accuracy and 0.76 error, which is acceptable, considering the simple architecture that we have.

Second experiment is trained on HPD train set and tested on HD test set. This experiment is aimed to evaluate the contribution of printed text as shown in Figure 5 we got 0.85% accuracy and 0.61 error.

Third experiment is trained on HSD train set and tested on HD test set. This experiment is aimed to show the contribution of synthesized text. As noticed from Figure 5 we got 0.835% accuracy and 0.6 error.

In addition; comparing the CNN behavior on HPD and HSD datasets we investigate the effectiveness of our dataset extension methods. These results lead us to conclude that the extension methods we used improves the prediction performance of the CNN, and does not cause overfitting.

VI. Conclusion

In this article we present two approaches to extend the dataset of Arabic subwords, and use them to train a CNN. First is extending a handwritten text image dataset using automatic synthesis. Second is generating printed text image dataset using Naskh font and degrading it to be similar to Arabic historical document. According to the results of three experiments we can conclude that using the extended dataset enhances the performance of the CNN, and does not lead to overfitting. Although we used a simple architecture for the CNN, we got encouraging results. To enhance the performance of the CNN we can build a more complicated architecture however, we will leave this for future work.

Acknowledgment

We thank Bilal Abu Saad for help with data and code. This work was partially supported by the Frankel Center for Computer Science at Ben-Gurion University of the Negev.

References

[1] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2. IEEE, 2003, pp. II–521.

[2] T. M. Rath and M. Raghavan, "Word spotting for historical documents," *International Journal of Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 139–152, 2007.

[3] T. M. Rath and R. Manmatha, "Features for word spotting in historical manuscripts," in *Document Analysis and Recognition Proceedings*. IEEE, 2003, pp. 218–222.

[4] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 3304–3308.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[6] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 440–445.

[7] M. Kassis and J. El-Sana, "Automatic synthesis of historical arabic text for word-spotting," in *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*. IEEE, 2016, pp. 239–244.

[8] S. A. ALshebeili, A. A.-F. Nabawi, and S. A. Mahmoud, "Arabic character recognition using 1-d slices of the character spectrum," *Signal Processing*, vol. 56, no. 1, pp. 59–75, 1997.

[9] T. S. El-Sheikh and R. M. Guindi, "Automatic recognition of isolated arabic characters," *Signal processing*, vol. 14, no. 2, pp. 177–184, 1988.

[10] S. A. Mahmoud, "Arabic character recognition using fourier descriptors and character contour encoding," *Pattern Recognition*, vol. 27, no. 6, pp. 815–824, 1994.

[11] N. Mezghani, A. Mitiche, and M. Cheriet, "On-line recognition of handwritten arabic characters using a kohonen neural network," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*. IEEE, 2002, pp. 490–495.

[12] N. Mezghani, A. Mitiche, and C. Mohamed, "Bayes classification of online arabic characters by gibbs modeling of class conditional densities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1121–1131, 2008.

[13] H. Almuallim and S. Yamaguchi, "A method of recognition of arabic cursive handwriting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 5, pp. 715–722, 1987.

[14] A. Amin, A. Kaced, J. Haton, and R. Mohr, "Handwritten arabic character recognition by the irac system," in *Proc. 5th Int. Conf. Pattern Recognition*, 1980, pp. 729–731.

[15] A. Amin and J. F. Mari, "Machine recognition and correction of printed arabic text," *IEEE Transactions on systems, man, and cybernetics*, vol. 19, no. 5, pp. 1300–1306, 1989.

[16] K. El Gowely, O. El Dessouki, and A. Nazif, "Multi-phase recognition of multifont photoscript arabic text," in *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, vol. 1. IEEE, 1990, pp. 700–702.

[17] T. Sari, L. Souici, and M. Sellami, "Off-line handwritten arabic character segmentation algorithm: Acsa," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*. IEEE, 2002, pp. 452–457.

[18] S. S. Maddouri and H. Amiri, "Combination of local and global vision modelling for arabic handwritten words recognition," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*. IEEE, 2002, pp. 128–135.

[19] R. Saabni and J. El-Sana, "Hierarchical on-line arabic handwriting recognition," in *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ser. ICDAR '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 867–871.

[20] S. S. Kuo and O. E. Agazzi, "Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 8, pp. 842–848, 1994.

[21] T. Rath, S. Kane, A. L. and. Partridge, and R. Manmatha, "Indexing for a digital library of george washingtons manuscripts: A study of word matching techniques," *CIIR Technical Report, University of Massachusetts Amherst.*, 2002.

[22] Y. Lu and C. L. Tan, "Word spotting in chinese document images without layout analysis," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3, 11-15 Aug. 2002, pp. 57–60vol.3.

[23] J. L. Rothfeder, S. Feng, and T. M. Rath, "Using corner feature correspondences to rank word images by similarity," *Computer Vision and Pattern Recognition Workshop*, vol. 3, p. 30, 2003.

[24] D. J. A. Bhardwaj and V. Govindaraju., "Script independent word spotting in multilingual documents," in *in 2nd Intl Workshop on Cross Lingual Information Access*, 2008, p. 4854.

[25] Y. Leydier, F. Lebourgeois, and H. Emptoz, "Text search for medieval manuscript images," *Pattern Recognition.*, vol. 40, no. 12, pp. 3552–3567, 2007.

[26] A. F. S. T. Adamek, N. E. Connor, "Word matching using single closed contours for indexing historical documents," *Journal on Document Analysis and Recognition*, vol. 9, no. 2, p. 153165, 2007.

[27] R. Saabni and J. El-Sana, "Word spotting for handwritten documents using chamfer distance and dynamic time warping," in *Document Recognition and Retrieval XVIII*, 2011.

[28] M. Rusinol, D. Aldavert, R. Toledo, and J. Lladós, "Browsing heterogeneous document collections by a segmentation-free word spotting method," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 63–67.

[29] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Efficient Exemplar Word Spotting," in *British Machine Vision Conference*, 2012, pp. 67.1–67.11.

[30] A. Kovalchuk, L. Wolf, and N. Dershowitz, "A simple and fast word spotting method," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 3–8.

[31] S. Fiel and R. Sablatnig, "Writer retrieval and writer identification using local features," in *In Proc. 10th International Workshop on Document Analysis Systems*, March 2012, pp. 145–149.

[32] V. Dovgalecs, A. Burnett, P. Tranouez, S. Nicolas, and L. Heutte, "Spot It! Finding Words and Patterns in Historical Documents," in *12th International Conference on Document Analysis and Recognition*, 2013, pp. 1039–1043.

[33] R. Shekhar and C. Jawahar, "Word image retrieval using bag of visual words," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, 2012, pp. 297–301.

[34] J. Lladós, M. Rusinol, A. Fornés, D. Fernández, and A. Dutta, "On the influence of word representations for handwritten word spotting in historical documents," *International journal of pattern recognition and artificial intelligence*, vol. 26, no. 05, p. 1263002, 2012.

[35] K. Zagoris, I. Pratikakis, and B. Gatos, "Segmentation-based historical handwritten word spotting using document-specific local features," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 9–14.

[36] I. Rabaev, I. Dinstein, J. El-Sana, and K. Kedem, "Segmentation-free keyword retrieval in historical document images," in *International Conference Image Analysis and Recognition*. Springer, 2014, pp. 369–378.
[37] V. M. L. at Ben-Gurion University of the Negev, "Arabic historical documents dataset," December 2016. [Online]. Available: http:www.cs.bgu.ac.il/vml
[38] F. Chollet, "keras," https://github.com/fchollet/keras, 2015.
[39] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688
[40] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.