

# Towards the Machine Reading of Arabic Calligraphy: A Letters Dataset and Corresponding Corpus of Text

Seetah ALSalamah, Ross King

Department of Computer Science

University of Manchester

Manchester, UK

[seetah.alsalamah@gmail.com](mailto:seetah.alsalamah@gmail.com); [ross.king@manchester.ac.uk](mailto:ross.king@manchester.ac.uk)

**Abstract**— Arabic calligraphy is one of the great art forms of the world. It displays Arabic phrases, commonly taken from the Holy Quran, in beautiful two-dimensional form. The use of two dimensions, and the interweaving of letters and words makes reading a far greater challenge for Artificial Intelligence (AI) than reading standard printed or hand-written Arabic. To approach this challenge, we have constructed a dataset of Arabic calligraphic letters, along with a corresponding corpus of phrases and quotes. The letters dataset contains a total of 3,467 images for 32 various categories of Arabic calligraphic-type letters. The associated text corpus contains 544 unique quoted phrases. These data were collected from various open sources on the web, and include examples from several Arabic calligraphic styles. We have also undertaken both an explorative statistical analysis of this data, and initial machine learning investigations. These analyses suggest that combining knowledge of a limited variety of Arabic calligraphy texts, with a successful machine will be sufficient for the machine reading of forms of Arabic calligraphy.

**Keywords**— Arabic language, corpora, pattern recognition, Arabic dataset, calligraphy.

## I. INTRODUCTION

Arabic calligraphy combines the arts of drawing and writing, producing a unique form of textual art. The art originated when Muslims started writing and documenting the Holy Quran [1]. Since then, Arabic calligraphy has been expanded and diversified in styles, producing one of the world's great art forms [2]. This art also holds much historical information, as Arabic calligraphy is used in much of Islamic art and architectural design [3]; therefore, such texts are a very valuable and rich resource for data. There is therefore a practical need to automate the reading of Arabic calligraphy. However, little research has been done in this area, and few resources exist.

The first research in this area used Liner Discernment Analysis (LDA) as a means of feature extraction from these documents [4]. This research focussed on extracting only three Arabic letters (aleph, lam, ain) and compared K-nearest neighbour with Naive Bayes classification obtaining excellent results. Back-propagation neural network classification was used in [5] to identify the font type of Arabic calligraphy. This study applied image binarisation with edge direction matrices for future extraction, and obtained 43.7% recognition accuracy. The novel approach of Triangle Model feature extraction was used in [3].

Distance-based methods were then used to compare images. Applying Multi-Layer Perceptron and Random Forests as classifiers, respectively gave average accuracy of 50% and 65%.

## II. ARABIC LANGUAGE SPECIFICATION:

The machine reading of Arabic presents many interesting challenges [5]. Arabic has 28 main letters, and is written from right to left. The letters are very sensitive to the use of discrete marks to distinguish between them. It is common to have three letters with the same body form, but distinguished with dots or marks placed above or under them (for example, ح خ ج). Arabic letters are also represented by different forms according to their position in the word: start, middle, last, or isolated. This is not the case for the vowel letters (ا اذ ز و), which have no start or middle location forms because they split words apart as they are not connected to any letters on the LHS (left hand side).

On top of these challenges, the machine reading of Arabic calligraphy has many additional interesting challenges: the variety of styles, the extra level of cursive forms, the interweaving of letters and words, rotations and intersections, etc. Figure 1 show the difference between Arabic text types.

Given all these difficulties for the machine reading of Arabic calligraphy, how is it possible to envisage an AI system that could work? The answer is the restricted text types that use Arabic calligraphy. Arabic calligraphy is not used to write general Arabic texts, but rather to glorify text from the Holy Quran. This restriction greatly simplifies the task and makes it feasible. In Bayesian terms, we know a lot about the prior distribution of calligraphic texts. Through extracting from each text phrase, a bag of letters can be obtained to simplify the indexing of the suggested answer. By comparing the result of letter spotting with the probable bag of letters, the correct text for the selected image can be found.



Figure 1: The difference between Arabic text types in the writing of one word (al-Arabiya).

### III. GENERATING THE DATA

The source of the data generation is a set of 1,000 Arabic calligraphy images. This set was collected from various Arab and Islamic open sources on the web. One of the main sources of the collected data is [6]. This website was established in 2012 to encourage the manipulation of this type of Arabic text for general non-commercial use. We selected the examples to include a variety of distinctive styles and texts. Figure 2 shows four examples from the collected dataset, which is the source of the generation process.

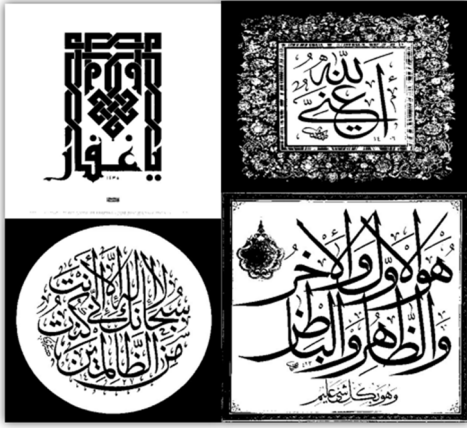


Figure 2: Examples of distinctive styles of Arabic calligraphy resources.

#### A. Creating the letter images dataset

The 28 main Arabic letters were included in the database with the addition of four other letters, giving 32 Arabic letters. The four extra letters were included because of their high frequency of use within the texts. These are the special format teeh marbuta (ة), alef maksura (ى) and the name of God (lellah, الله), along with the short vowel, hamza (ء). We manually extracted the letters from the calligraphic images. The aim was to form a dataset of Arabic letters, including their positional versions in the images, and to associate these with the Arabic text. For each letter we collected as many various forms and styles as we could find.

This annotation involved extracting letter images from the Arabic calligraphy images. The ImageJ<sup>1</sup> program was used for this process of manual annotation of the calligraphy. The process goes through different steps to enable extraction of the letters as images, saving each to its corresponding category. At the same time, the order of the letters and the extracted text were recorded. In addition, it was recorded which letter samples were extracted from which images. Working in parallel, the annotations from ImageJ and all of the letter sequences were recorded along with the number of the sample, to give the annotation of the letters more utility.

<sup>1</sup> ImageJ is an open source software package for image processing – <https://imagej.net/ImageJ>.

For each input image, the starting point was to convert it to binary (black and white). This was done to make it easier to remove noise, enhance contrast and manually extract letters as images. In addition, the threshold method was used, with a threshold of 50% and pixel radius of 2.0, to remove bright outliers and make the text clearer. Figure 3 shows an example of the letter image extraction from one input image resulting in 14 different Arabic calligraphic style letters in grayscale JPG format. Some letters, as shown in Figure 3, are extracted with some separation on the body form of the letter. This helps to simplify the detection and the spotting of such letters in different images when letters intersect with each other, as is common in the Arabic calligraphy domain.

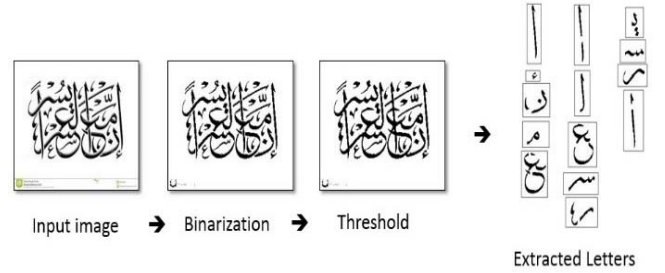


Figure 3: The process of extracting letters as images using the ImageJ tool.

Since some common letters are more frequent than others, we decided to obtain a stratified sample of letters, to provide a minimum of 100 samples for each letter. (The aim is to later use the empirical frequency of each letter in our corpus as a prior distribution.) This resulted in a total of 3,200 Arabic calligraphic letter images. Figure 4 shows different samples for different Arabic letters with various forms and styles. In this situation, all possible forms of the letters (initial, middle, last, isolated) are saved under the same class name. This is to enable the machine to later learn and spot the different forms and styles of every single letter.

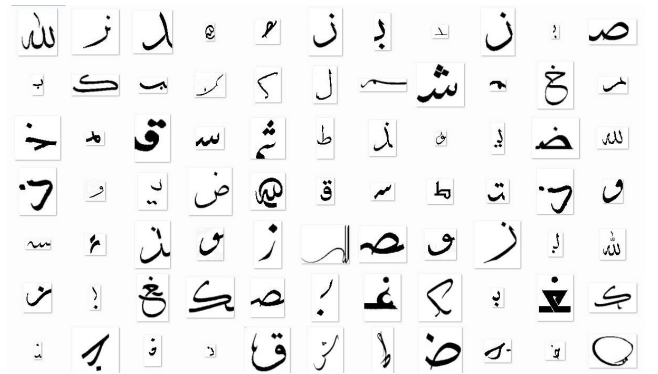


Figure 4: Different samples from the generated Arabic calligraphy letters dataset.

### B. Creating the corpus of text from quotes and phrases

The study focused on checking the source texts for Arabic calligraphy images to support the results of the machine reading. To start with, each image was manually checked and the text was extracted to produce a collection of 1,000 phrases of text or quotes from the images. We then ran descriptive statistics on the texts. The results showed that, from 1,000 images there were only 544 unique quotes. Table 1 shows statistics for the resulting corpus of phrases extracted from unique quotes. Table 2 shows the ten most frequent letters, words, and two-word sequences.

**Table 1 : Statistical summary for the phrases from the quotes.**

Number of Letters	17,648
Number of Words	4,252
Number of Sentences	544
Unique Words	1,237
Short Words ( $\leq 3$ letters)	1,569
Long Words ( $\geq 7$ letters)	285
Average Word Length / letters	4
Average Sentence Length / words	8.1

**Table 2: The ten most frequent letters, words, two-word sequences and three-word sequences found in the corpus of quotes given as percentages.**

Let	%	1 word	%	2 words	%	3 words	%
ل	16	الله	8.9	الرحمن الرحيم	1.6	بسم الله الرحمن الرحيم	1.5
ا	16	الا	1.8	بسم الله	1.6	الله الرحمن الرحيم	1.5
م	7.7	من	1.8	الله الرحمن	1.5	لا إله الا	0.5
و	6.1	الرحيم	1.7	الله عنه	0.7	الله عليه وسلم	0.4
ن	6.1	الرحمن	1.7	إله الا	0.6	إله الله	0.4
ي	6	بسم	1.6	لا إله	0.6	الله الله	0.3
ه	6	لا	1.3	الله لا	0.5	صل الله عليه	0.3
ر	4.9	ولا	1.2	الله عليه	0.4	رضي الله عنه	0.3
ب	3.6	ان	0.9	عليه وسلم	0.4	رضي الله عنه	0.3
ع	3.2	محمد	0.9	رسول الله	0.4	الله لا إله	0.3

As the key difficulty in reading Arabic calligraphy is identifying the relationship between letters, which is provided in standard text by its one-dimensional form, we investigated the uniqueness of ‘bags of letters’ in our texts. The idea is that, if we could accurately identify the individual letters in a calligraphic image, how much would that help us to identify the actual text. We therefore counted the frequency of each Arabic letter in every quote. Table 3 shows an example of such a bag of letters from the quote (بسم الله الرحمن الرحيم). Each bag of letters is saved for its corresponding quote to be used later in a further relational model.

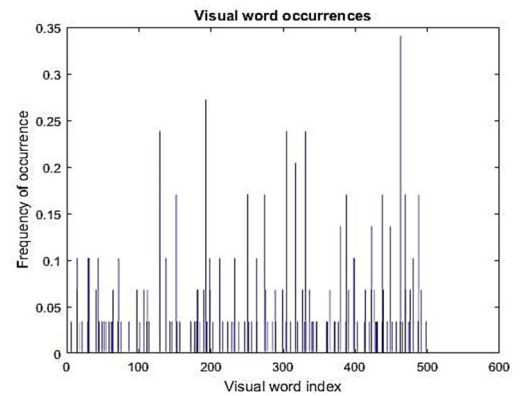
**Table 3: Example of a bag of letters for one quote.**

Quote	Letter	Occurrence
بسم الله الرحمن الرحيم	ل	4
	م	3
	ا	3
	ر	2
	ح	2
	ب	1
	س	1
	ه	1
	ن	1
	ي	1

### IV. APPLICATION OF MACHINE LEARNING TO LETTER RECOGNITION

Our manual annotation generated a set of examples for each letter class. The goal was to generate machine learning classifiers for each letter. We first normalised the images dataset to a fixed size of  $277 \times 277$  pixels. With 100 examples for each of the 32 Arabic letter categories, this gives a total of 3,200 different Arabic calligraphy images. We split the data into 70% for training, and 30% for testing.

We applied Support Vector Machines (SVM) using the polynomial kernel, with two feature extraction methods. These feature extraction methods were bag of features, and the Histogram of Oriented Gradients (HOG) features [7]. The two models use the same learning method but there is a different feature extraction for each of them. The first model used bag of features, or visual word features. For each image, the strongest Speeded Up Robust Features (SURF) were extracted. This feature extraction method was used to detect and describe local features for each image [8]. The bag of features was then constructed by reducing the number of extracted features to a fixed set to give a vocabulary of words describing each image. These features were encoded into a visual word histogram. The extracted features for each image were then indexed to a letter category. Figure 5 shows an example of extracted encoded bag of word features.



**Figure 5: The histogram of visual word rates**

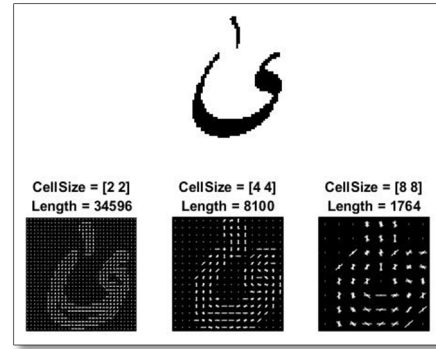
The second SVM model used HOG feature extraction. Figure 6 shows the different cell size features for the representation of the Arabic letter alef-maksura. Cell [8 8] has been selected as the cell size in the model since it represents the highest level of feature details. The results of these two models are shown in Table 4, with the second model outperforming the first by more than 3%.

**Table 4 : The result of Arabic Letter Classification Learnable model.**

Learning model	Feature extraction	Accuracy rate
SVM	Bag of SURF	57%
SVM	HOG	60%

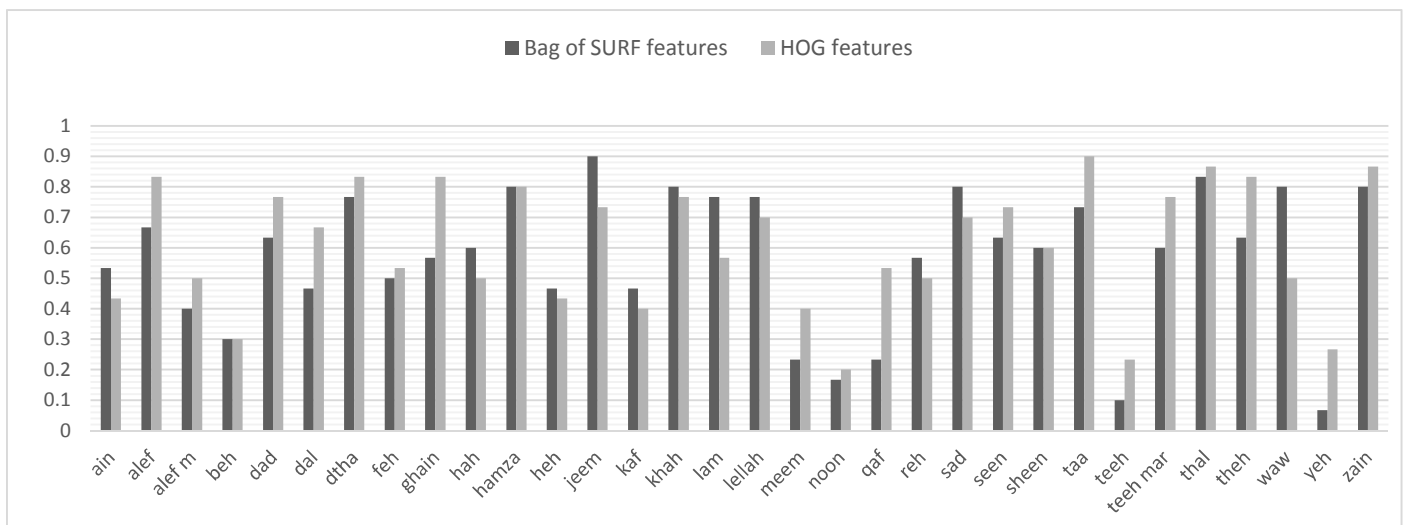
The results for accuracy of prediction for each letter class with both SVM models are shown in Figure 7. The second model, the HOG features, gave a better description of the letter features than the bag of SURF features, as it outperformed the first model for those letters with more details and challenges. It seems that the HOG feature is better at describing the letters with the dots and extra descriptors as this model had better predictions for 18 letters and had the same results as the first model for three letters, beh ب, hamza ء, sheen ش. Meanwhile, the first model had better predictions than the second model for only 11 letter classes. From these results, we have an overview of the recognition of complex letters, which need further simplification methods or procedures to make them easier for the machine to read. The results in Figure 7 show that the letters, beh ب, meem م, noon ن, teeh ت, yeh ي, have low prediction rates in both models compared with the other letters. This is due to the complexity of these letters forms or their similarity to other letter classes.

In a further test, ten different new images were used to examine the letter classification of the model. Table 5 shows the ten different test images and the results of the system's letter classification. From the ten tested input images, the recognition by the letter classification system can be clearly observed and ana-



**Figure 6: HOG feature representation for alef-maksura in different cell size.**

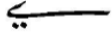



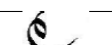















lysed. The confusion between the letter forms in Arabic calligraphic styles results in the incorrect classification of the input images. The sixth column in Table 5 shows the cause of this incorrect classification to result from the similarity of forms in the different calligraphic letter classes. There are only three wrong classifications (examples 2, 3 and 6), as these are simple and clear from the additional cursive style. From the ten inputs tested, there is an accuracy of 7/10, or 70%, showing good behaviour in distinguishing between these letters. Moreover, the result for example 9 is not similar to the actual letter form, as it is rotated, yet it is successfully predicted. On the other hand, there appears to be no reason for the misclassified examples because these results are not related to the input image, the forms of the letter bodies or the sharing of a common style. These results help in analysing and evaluating how learnable models will perform with such types of data. These experiments give a good starting point for understanding how machines could distinguish between the image datasets for the different Arabic calligraphic styles, as well as how they could perform with the different forms and styles of the same letter if we combine them together. This is just the start of the process to enable machine reading of Arabic calligraphic images.



**Figure 7: Result of SVM classification for each letter category with the different feature extraction methods.**



**Table 5 : Ten examples of testing the classification system with the corresponding results for each input.**

Num	Input Letter	Actual input image	Classified category	Classified letter image
1	'yah'		'yah' ✓	
2	'alef'		'yah' ✗	
3	'meem'		'alef-m' ✗	
4	'reh'		'reh' ✓	
5	'heh'		'heh' ✓	
6	'noon'		'meem' ✗	
7	'noon'		'noon' ✓	
8	'seen'		'seen' ✓	
9	'feh'		'feh' ✓	
10	'dal'		'dal' ✓	

## V. DISCUSSION AND CONCLUSIONS

We plan to extend our dataset and corpus by more than doubling the number of examples. This will require the extraction of more images of letters, and the inclusion of examples formed by rotating the forms in different directions and through other forms of image processing manipulation. This process will require care to ensure that no bias enters the process, and a significant proportion of the samples will be retained as a final test set.

We have also extended our investigations into the development of machine learning methods for letter, word, and phrase recognition. Integration of predictions at these distinct levels, along with a priori knowledge about the distribution of Arabic texts used in calligraphy will also present an interesting research challenge.

The reading of Arabic calligraphy presents an interesting challenge to AI systems, and one in which there has been little previous work. We have presented our initial results in this area, and we have collected an initial letter database and associated text corpus. When completed, we will make our datasets publicly available. Research data is rare in the field of the machine reading of Arabic, and this has restricted progress in the research area. We therefore hope that our dataset will contribute to further computational studies of Arabic calligraphy. Our initial results suggest that the constraints on the texts that use Arabic calligraphy, combined with reasonable machine success, will be sufficient for machine reading of some forms of Arabic calligraphy.

## VI. BIBLIOGRAPHY

- [1] N. Dershowitz and A. Rosenberg, "Arabic Character Recognition," in *Language, Culture, Computation. Computing-Theory and Technology*, vol. 31, Springer, 2014, pp. 584-602.
- [2] E. Ataer and P. Duygulu, "Retrieval of Ottoman documents," in *Proceedings of the 8th ACM international workshop on multimedia information retrieval*, California, USA, 2006.
- [3] M. S. Azmi, K. Omar, M. F. Nasrudin, A. K. Muda, A. Abdullah and K. W. M. Ghazali, "Features extraction of Arabic calligraphy using extended triangle model for digital jawi paleography analysis," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 5, pp. 696-703, 2013.
- [4] I. Bar-Yosef, I. Beckman, K. Kedem and I. Dinstein, "Binarization, character extraction, and writer identification of historical Hebrew calligraphy documents," *Springer*, vol. 9, no. 2, pp. 89-99, 2007.
- [5] B. Bataineh, S. N. H. S. Abdullah and K. Omar, "Arabic calligraphy recognition based on binarization methods and degraded images," in *Pattern Analysis and Intelligent Robotics (ICPAIR)*, Malaysia, 2011.
- [6] P. G. b. M. b. Talal, "Free Islamic Calligraphy," the Prince Ghazi Trust for Qur'anic Thought, 2012. [Online]. Available: <http://freeislamiccalligraphy.com/>. [Accessed 2017].
- [7] X. Yang, C. Zhang and Y. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in *Proceedings of the 20th ACM international conference on Multimedia*, Nara, Japan, 2012.
- [8] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008.
- [9] B. Bataineh, S. N. H. S. Abdullah and K. Omar, "Generating an Arabic Calligraphy Text Blocks for Global Texture Analysis," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 1, no. 2, pp. 150-155, 2011.
- [10] A. A. Aburas and S. A. Rehiel, "JPEG for Arabic Handwritten Character Recognition: Add a Dimension of Application," Malaysia, INTECH Open Access Publisher, 2008, p. 472.
- [11] Y. Zhu, T. Tan and Y. Wang, "Font recognition based on global texture analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1192-1200, 2001.
- [12] Z. Ahmad, J. K. Orakzai and I. Shamsher, "Urdu compound character recognition using feed forward neural networks," in *2nd IEEE International Conference on Computer Science and Information Technology*, 2009.
- [13] O. Al-Jarrah, S. Al-Kiswani, B. Al-Gharaibeh, M. Fraiwan and H. Khasawneh, "A new algorithm for Arabic optical character recognition," *WSEAS Transactions on Information Science and Applications*, vol. 3, no. 4, pp. 832-845, 2006.
- [14] H. A. Al-Muhtaseba, S. A. Mahmouda and R. S. Qahwajib, "Recognition of off-line printed Arabic text using Hidden Markov Models," *Signal Processing*, vol. 88, no. 12, pp. 2902-2912, 2008.
- [15] M. A. Abdullah, L. M. Al-Harigy and H. H. Al-Fraidi, "Off-line Arabic handwriting character recognition using word segmentation," *arXiv preprint arXiv:1206.1518*, 2012.