Using Prior Knowledge to Facilitate Computational Reading of Arabic Calligraphy

Seetah ALSalamah, Riza Batista-Navarro(✉), and Ross D. King(✉)

Department of Computer Science, University of Manchester, Manchester, UK
Seetah.alsalamah@gmail.com,
riza.batista@manchester.ac.uk, rdking@turing.ac.uk

Abstract. Arabic calligraphy (AC) is central to Arabic cultural heritage and has been used since its introduction, with the first writing of the Holy Quran, up until the present. It is famous for the artistic and complicated ways that letters and words interweave and intertwine to express textual statements – usually quotations from the Quran. These specifications make it probably the hardest of all human writing systems to read. Here, we introduce the challenge of reading Arabic calligraphy using artificial intelligence (AI), a challenge that combines image processing and understanding of texts. We have collected a corpus of 1000 AC images along with annotated quotations from the Quran, pre-processing the images and identifying individual letters using detection methods based on maximally stable extremal regions (MSERs) and sliding windows (SWs). We then collect the identified letters to form bags of extracted letters (BOLs). These BOLs are then used to search for possible quotation from the corpus. Our results show that MSERs outperforms SWs in letter detection. Furthermore, BOL-matching is better than word generation in predicting the correct quotation, with the correct answer found in the list of 10 topmost matches for more than 74% of the 388 test examples.

Keywords: Computational reading · Arabic calligraphy · Pattern recognition · Natural language processing

1 Introduction

Artificial intelligence (AI) has recently made rapid advances in image analysis and understanding of texts, owing to the advent of massive annotated image databases and new developments in machine learning [1]. Arabic is one of the most widely spoken languages in the world. Written Arabic has a number of features that complicate its computational reading: (1) Arabic letters can assume a variety of shapes depending on their position within a word, and (2) some letters come in families that are similar in shape but are distinguished by markers, i.e., dots, that appear above or below the main shape [2].

Arabic calligraphy (AC) is a famous element of Islamic cultural heritage, with an artistic practice based upon the beauty of intersecting script [3]. Arabic quotations, often from the Quran, are beautifully written in AC. In part, because of Islamic an iconism, AC has been a focus of Arabic artistic achievement for over 1400 years,

H. Yin et al. (Eds.): IDEAL 2019, LNCS 11872, pp. 293–304, 2019.
https://doi.org/10.1007/978-3-030-33617-2_30

and many different styles of AC have been developed [4]. To conform to the artistic style of AC, letters are written in a manner where they appear more cursive, are arranged to intersect, interleave, and interweave. Additionally, words are often reordered [5]. These changes in shape and rearrangements of letters and words make AC difficult to read, even for native Arabic readers [4].

Moreover, there are more than 50 different AC styles; the most commonly used are Naskh, Thuluth, Diwani, Nastaliq, Riqa and Kufi [6]. Each AC style has its individual specifications, with different levels of overlap, letter rotation and cursiveness, in addition to different degrees to which marks are drawn around the text. Figure 1 shows examples of six different AC artistic styles expressing the same quotation. These show the variety of representations of the text in stylistic drawing arts, adding to the existing challenge presented by Arabic texts [7].

In this paper, we propose a new method to read the drawn text in the AC images. Our method works at the letter level, to allow for extracting letters from the intersections and interweaving of text in the image. It relies on the availability of calligraphy quotations to simplify the mapping from extracted letters to the target quote.

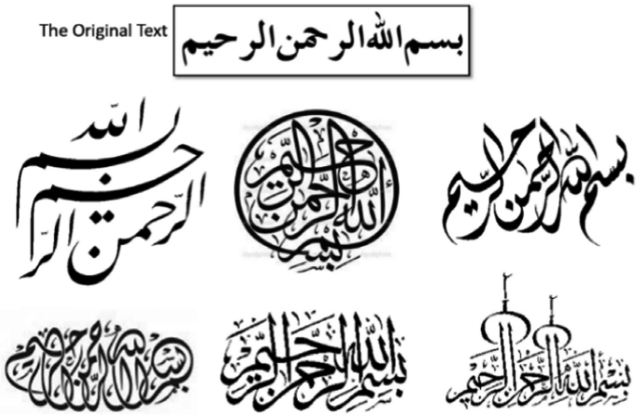

Fig. 1. Original Arabic text (above) and its representation in six different styles of AC, is the famous Arabic phrase translated as "In the name of God, most Gracious, most Compassionate".

2 Related Work

Digitising old documents is a trend in the online era, in which everything is now placed and allocated in order to be available in various media, to be manipulated, translated or learned. This mainly focusses on detecting text from source images; this has been recognised in many types of research that hasbeen conducted in different languages and working on various types of text, whether handwritten or computer-based, or on styles of calligraphic art [8]. However, little of this research has handled calligraphy-based text, since the challenges in this type of text mean dealing with it as shapes rather than letters [9]. On the other hand, there are impressive results from Chinese text

recognition, which shares with Arabic the complexity of intersecting letters or characters [10]. However, compared to Arabic letters, which have different shapes according to their location in the word, Chinese characters have constant shapes.

Ye and Doermann [11] summarise research on text detection and recognition in imagery by comparing the methods used and the challenges found in reading text from coloured images in different languages (English, French, Chinese and Korean). They find that, for more than ten different approaches, the results are not very good when stroke detection and word spotting approaches are applied; even with the best end-to-end methods, the final reading accuracy is less than 50%. They cover the remaining problems that reduce the accuracy of reading text from images in three main points as follows: (1) the requirement for a large-vocabulary corpus to obtain the best selection of all probable words; (2) the need to continue improving the correct detection of characters from coloured images; and (3) challenges of building a joint model that can handle multiple languages.

For the Arabic language, previous work done on text recognition achieved more than 90% accuracy [12] on text handwritten in standard Arabic. However, AC cannot be manipulated like standard Arabic text since the intersection of letters needs special segmentation methods [13]. The randomness and challenges in the different styles of AC [4] make its reading difficult, even for those who understand its sophisticated and intersecting styles. Previously reported work focussed on analysing various types of AC [3] and recognising AC text in historical documents [13–15], showing that computational analysis of AC is difficult. As a result, computational reading of AC has been underexplored, and there is a lack of supporting computational resources [4]. To address this gap, we have created our particular dataset to enable the process of reading AC. The initially collected datasets have been published in [16] and will be publicly available after finalising the data collection.

3 Arabic Calligraphy Datasets

To support the development of our computational reading methods, we constructed datasets drawn from AC images. Firstly, more than 1000 AC images were collected from open source websites, mostly from the Free Islamic Calligraphy online platform [17]. These images were then manually annotated to extract: a dataset of letter images and a corpus with the corresponding text. The manual annotation was done by segmenting AC letters and storing them as separate, individual images and, at the same time, by recording the drawn text as a single quotation in the corpus. Figure 2 shows an example of the segmentation of letters into separate images from calligraphy image no.99 and recording of the quotation in the calligraphy corpus.

The resulting A Cletter image dataset contains more than 3200 letter images, with 100 samples for each letter. There are 32 different letter categories, 28 for the main Arabic letters along with four special letters frequently used in the AC context. These letters are the name of God ﷲ, *teehmarbuta* (ة), *alefmaksura* (ى) and the short vowel, *hamza*(ء). All of these images have been saved in their original size upon extraction, with black letters against a white background.

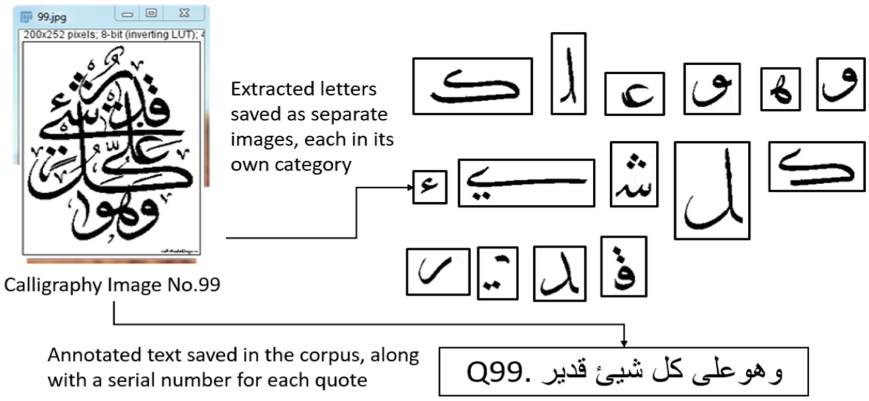

Fig. 2. Extraction of the letter image dataset and the corresponding corpus from Arabic calligraphy images.

Meanwhile, the calligraphy corpus contains more than 528 different quotations annotated from the calligraphy images. The final annotations were extracted from 1000 images resulting in only 528 unique quotations, with the rest being duplicates written in a different style. The number of words in the quotations ranges from one word to a maximum of 285 words, with an average of four words per quotation. Analysis of the most frequent single character, word and sets of *n*-words has been conducted to explore common terms in the AC domain. All of the annotation procedures have been approved by a team of three native Arabic readers. Each member of the team completed his/her annotations separately; the final corpus is the result of selecting only matching annotations.

4 Methodology

We decomposed the computational reading of AC images into a series of steps, shown in Fig. 3. The input image is first pre-processed to remove noise and then enhanced by filtering and sharpening. The second step involves the segmentation detection of letters, not words, are extracted to attempt to resolve the disordering of text in AC. Two different methods were used in the detection of letters: maximally stable extremal regions (MSER) and sliding window (SW). With both of these methods, the resulting extracted letters are evaluated separately. In the next stage, the extracted labelled letters are passed to the quotation matching stage to attempt to map them to the probable quotation. The result for each set of extracted letters is a list of likely quotations ranked most likely to least likely. Each of these stages is explained in more detail below.

4.1 Image Pre-processing

In this stage, image enhancement and correction techniques are used to try to improve image quality. The final results are very sensitive to this stage, as AC images generally

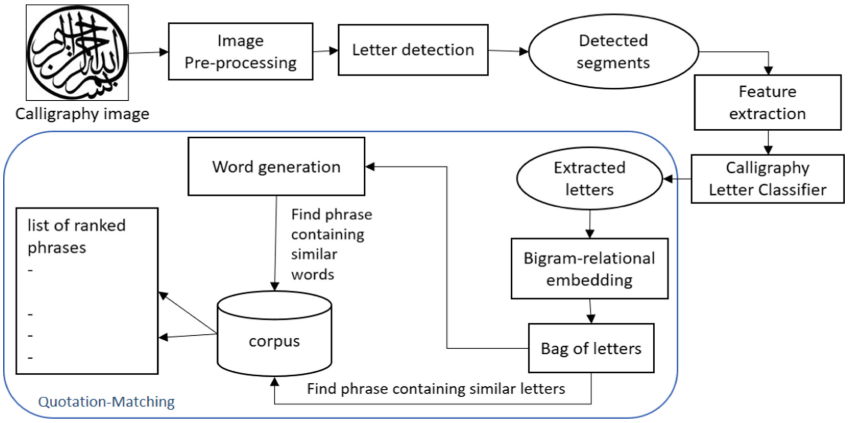

Fig. 3. Diagram depicting our approach to the computational reading of AC, starting from a given input image and ending with the list of probable quotations.

contain significant noise due to their different colours and styles. First, for each image the sharpness of the letter edges is increased, making the letters as clear as possible, capturing as many of the small details as possible. This improves the detail of the image edges and sharpens them. We used the standard deviation of a Gaussian filter to adjust the sharpness of the edges [18]. Secondly, binarisation, also called thresholding, is applied, focussing mainly on removing as much as possible the remaining noise from the image [19]. Finally, small noisy objects are removed; the noise, in this case, was contributed by the artist and the surrounding the text. These are removed from the image by identifying all objects in the image and removing those of a specific size. Based on our manual inspection of 100 images, the smallest object contained in an image typically in the range of 50 to 70 pixels size. We thus eliminated the noise from our images by removing any object that is not bigger than this size. Figure 4 shows an example of an AC image and the results of removing small objects from it in the manner that we have just described. We show the results of using two values of a threshold t, i.e., 50 and 80 pixels, to illustrate the difference between removing objects of size 50 pixels and fewer, and removing objects of size 80 pixels and fewer. With $t = 50$ pixels, some noise remains, but the image is much cleaner than the original image. When $t = 80$ pixels, some of the dots in the letters are removed with the noise (dots in *noon* and *beh*), which is not desirable. This implies that removing noise is a delicate process that could affect the accuracy of letter detection.

4.2 Letter Detection and Recognition

Letter detection involves the extraction of letters from among all of the intersecting and connected shapes within a given AC image. This needs to be carried out without contaminating the letter with noise while selecting the correct location of any related marks (e.g., dots) if they are present. Two detection methods were employed to locate

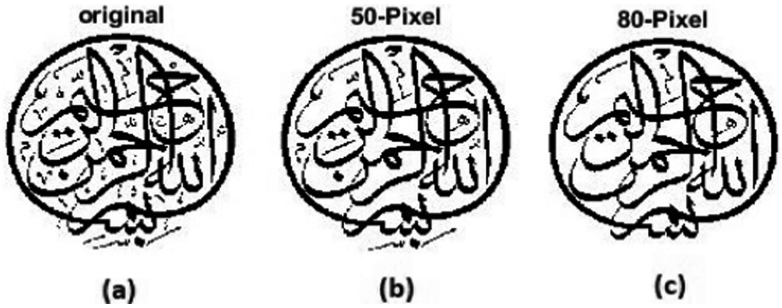

Fig. 4. Example AC image (a) and the results of removing any object whose size is 50 pixels or less (b) and of removing any object whose size is 80 pixels or less (c).

the different letters in an image: maximally stable extremal regions (MSERs) and sliding windows(SWs), which we describe next.

Maximally Stable Extremal Regions (MSERs). This process segments images into regions (corresponding to objects) depending on a grey-level image threshold [20]. The image is scanned according to a threshold, grouping the parts of objects that share the same grey level within their boundaries as one object. Figure 5 shows an example of an AC image and the MSERs detected, shown as coloured segments.

Fig. 5. The MSERs detected in an AC image, shown as coloured segments.

Sliding Windows (SWs). This process uses a rectangular region with a fixed size (64 × 64 pixels) as a window for object detection [21]. This window is used to scan the whole image, moving from left to right and row by row, with the pixels within the window checked continuously for any object that might be contained. The window size is 64 × 64 pixels, similar to the image size that our AC letter classifier was trained on [16]. Every time the window moves to a new position, the features are extracted and sent to the classifier to check whether or not they relate to any letter category.

After an object is identified, it is stored as a separate image and given as input to a classifier for letter recognition. Our AC letter classifier is underpinned by a model trained using support vector machines (SVMs) [16], on two different feature sets: a bag of visual speeded up robust features (SURF) and histogram of oriented gradient (HOG) features. Whereas the former is based on scale-invariant descriptors of image points of interest [22], the latter is counts occurrences of gradient orientation in

localised parts of an image [23]. Previous work has shown that the HOG features out perform SURF, with an accuracy of 60% compared with 57% [16]. Thus, the SVM classifier trained on HOG features was selected as our letter recogniser. The classifier was trained with 3200 different image samples for 32 letter categories. The process involved passing an entire calligraphy image through each of the detection methods (described above) and then, for each resulting object segment, having the classifier predict this object's letter category. At the end of this process, the letters are saved as a 'bag of letters': a vector whose elements correspond to the frequency with which each of the 31 Arabic letters of interest was recognised in the given image. The name of God ﷲ category is not placed in the vector as the representative letter (2 *lam,* 1 *hah*) will be separately counted.

4.3 Quotation Matching

It is worth noting that the calligraphic texts that we are trying to read computationally are known, as they are quotations from the Holy Quran and the Sunnah Hadith[1]. This significantly reduces the difficulty of the task and makes the reading of AC in images feasible. We approached quotation matching in two different ways. The first approach made use of a bag of letters (extracted in the previous step) in searching for text in the corpus of quotations. The second approach was based on the generation of all possible words from the extracted letters and then finding quotations containing these words. Meanwhile, for each quotation in the corpus, the frequency of each of the letters it contains is declared to facilitate the search process, described in more detail below.

Bag-of-Letters (BOL) Matching. Firstly, all of the extracted letters are used to form a bag-of-letters vector, which will be used in checking for similarity with the available quotations in the corpus. In filling in the values of this vector, we enforced rules drawn from prior knowledge on letter bigrams that appear most frequently in calligraphic quotations. Such rules were incorporated based on the intuition that accounting for known related letters could compensate for letters that object detection sometimes fails to extract [24]. Table 1 shows the 20 most frequent letter bigrams found in the calligraphy corpus, along with the details of their frequency of occurrence. These are translated into rules which define that if one of these bigram letters is detected (and counted), then the second letter is also counted. These bigrams are restricted in that they occur in only one direction, i.e., from right to left. Therefore, if the right-hand letter is found, then the next letter is added, otherwise there is no addition. This is a unidirectional rule and, for that reason, there is a difference between ال and لا (bigrams with the same letters but in a different order and with different frequencies), for example.

We now describe the process for matching quotations in the corpus. Firstly, the similarity between the bag of letters obtained for a given AC image Bag_{Si} and the bag of letters Bag_{Ti} for every quotation in the corpus, is measured based on the score n which is defined as:

[1] The Sunnah Hadith is the second most important textual source of knowledge in Islam after the Holy Quran. It is the report of the Prophet Muhammad's words or actions.

Table 1. The most frequent bigrams found in the calligraphy phrases corpus.

Serial	Bigram	Count	%	Serial	Bigram	Count	%
1	ال	701	6.51%	11	ول	85	0.79%
2	له	264	2.45%	12	حم	83	0.77%
3	لل	212	1.97%	13	ين	79	0.73%
4	لا	125	1.16%	14	يم	77	0.72%
5	لم	122	1.13%	15	لر	76	0.71%
6	وا	110	1.02%	16	لي	70	0.65%
7	من	101	0.94%	17	ما	68	0.63%
8	ان	99	0.92%	18	نا	56	0.52%
9	عل	90	0.84%	19	رب	55	0.51%
10	رح	88	0.82%	20	با	54	0.50%

$$n = \sum_{i \in 0..30}^{let} \begin{cases} 0, & if\ Bag_{Si} == Bag_{Ti} \\ 1, & otherwise \end{cases} \tag{1}$$

Here, i corresponds to the vector element position, whose value ranges from 0 to 30. Hence the two bags are compared element-per-element. If the vectors share the same value for a specific element (i.e., a letter), a score of 0 is returned; otherwise, the score returned is 1. At the end of this element-wise comparison, the value of n will hold the summation of scores for all elements. A smaller value of n thus implies a higher similarity between the input and quotation bags of words. We can then use n to induce a ranking amongst the quotations in the corpus.

Word Generation. In this step, all of the possible context-based words that can be built using the extracted letters, are generated. Given the set of extracted letters $Let_{1..m}$ (together with their related bigrams), possible context-based words containing the maximum number of extracted letters can be predicted according to the naïve Bayes probability, where m is the number of extracted letters:

$$P(W_{ar}|Let_{1..m}) = P(Let_{1..m}|W_{ar}) \times P(W_{ar}) \tag{2}$$

Then, by selecting quotations that contain two or more sequences of the generated words, we obtain a list of quotations ranked according to the number of generated words they contain.

5 Experiments

To evaluate our approach to the computational reading of AC images, we used a test set containing 388 different AC images. This set was labelled with ground truth quotations and contained a wide variety of different styles (Naskh, Thuluth, Diwani, Riqa and Kufi), in various text sizes and representations. Figure 6 shows some samples from the test.

Fig. 6. Samples from the AC image test set.

All 388 images were analysed by the whole sequence of processes described above to predict the quotations they contain. Both MSERs and SWs were employed in the detection phase. Results from both BOL matching and word generation were used in quotation matching. The following parameter values were used in the image pre-processing stage: all objects in the images of size 70 pixels or less were removed; images were binarised with a threshold of 0.4; and, image edges were sharpened with a radius of 2 and strength of 2. All of these parameters have been carefully chosen to optimise performance, based on the result of experiments with small samples of 15 different AC images not from the tested samples.

We evaluate the performance by calculating Top N accuracy [25], which is the percentage having at least one correct quotation in the top-N probable quotations list keeping the number of totals tested images in consideration. Figure 7 shows the Top-10 accuracy results from BOL matching comparing the two methods of detection, MSER and SW. MSER outperforms SW with more correct predictions at a lower level and in the general total. Most of the correct results are found in levels 6 to 10. Altogether, more than 75% for MSER and 68% for SW were correctly located in the top ten prediction levels.

Figure 8 shows the Top-6 accuracy results for word generation from the extracted letters for both methods of detection, MSER and SW. The N level refers to the number of words generated and found in the same predicted quotations with a minimum of 2 words (starting from level 2). In general, the performance of this detection method is inferior to that of BOL matching, as the correct results were only 23%, for SW, and 19%, for MSER, of the total image testing set used. Since SW involves moving around the image and generating more letters than MSER, it is described as having generated slightly more words than MSER.

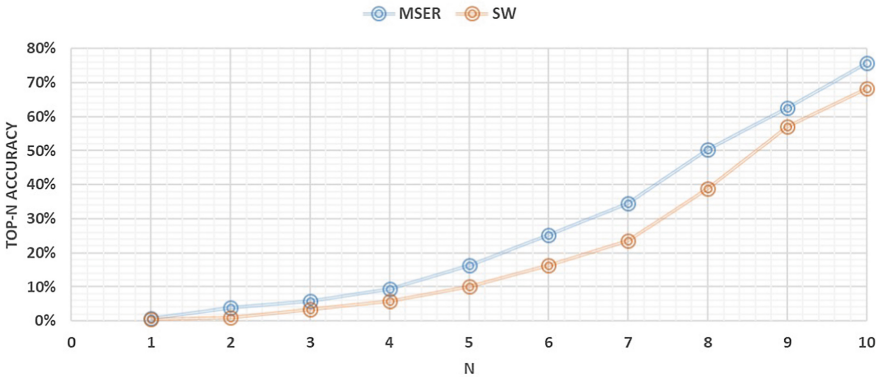

Fig. 7. Top-10 Accuracy results from BOL matching

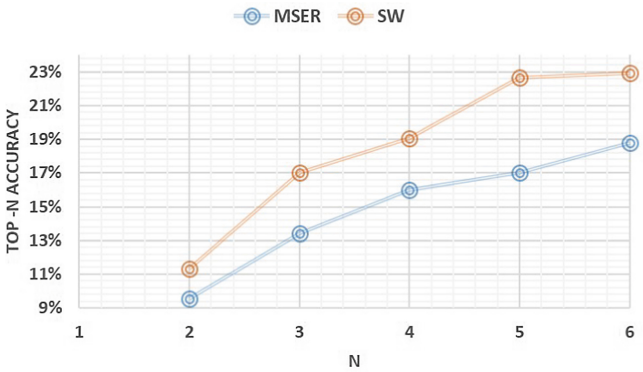

Fig. 8. Top-6 Accuracy results for word generation.

6 Conclusion and Future Work

In this paper, we have compared two methods of detection of AC images, MSER and SW, by using two different search techniques. The results show that BOL matching outperforms word generation from extracted letters, giving an accuracy for all of the top ten suggestions of 75% for MSER and 68% for SW. This is because, in this method, the corpus is directly searched for the bag-of-letter features to present possible suggestions for the text. The conclusion is that MSER gives more precise letter detection, while SW repeats and produces more letters. This helps with word generation, but not with BOL matching, as this is affected by any extra letters. The results so far are promising in terms of further improving the detection and reading of the text in these types of images. In future work, we plan to improve the quotation matching stage by the addition of more methods to map to the correct answer. If our calligraphy dataset could be significantly extended, this would enable the utilisation of state-of-the-art deep learning methods, which have demonstrated their great utility on image analysis problems [26].

Acknowledgement. SA would like to thank King Saud University for funding this research.

References

1. Sun, Y., Zhang, C., Huang, Z., Liu, J., Han, J., Ding, E.: TextNet: Irregular Text Reading from Images with an End-to-End Trainable Network, pp. 1–17 (2018)
2. Azmi, A., Alsaiari, A.: Arabic typography: a survey. Int. J. Electr. Comput. Sci. **9**(10), 16–22 (2010)
3. Bataineh, B., Norul, S., Sheikh, H., Omar, K.: Generating an arabic calligraphy text blocks for global texture analysis. Int. J. Adv. Sci. Eng. Inf. Technol. **1**, 150–155 (2011)
4. Saberi, A., et al.: Evaluating the legibility of decorative Arabic scripts for Sultan Alauddin mosque using an enhanced soft-computing hybrid algorithm. Comput. Human Behav. **55**, 127–144 (2016)
5. Mohamed, N.A., Youssef, K.T.: Arts and design studies utilization of arabic calligraphy to promote the arabic identity in packaging designs. Arts Des. Stud. **19**, 35–49 (2014)
6. Bataineh, B., Abdullah, S.N.H.S., Omar, K.: A novel statistical feature extraction method for textual images: Optical font recognition. Expert Syst. Appl. **39**(5), 5470–5477 (2012)
7. Aburas, A.A., Gumah, M.E.: Arabic handwriting recognition : challenges and solutions electrical and computer engineering dept international islamic university malaysia department of information technology, university technology PETRONAS 2. Pervious related Research Work. In: International Symposium on Information Technology, pp. 1–6 (2008)
8. Bhowmik, S., Sarkar, R., Nasipuri, M., Doermann, D.: Text and non-text separation in offline document images: a survey. Int. J. Doc. Anal. Recognit. **21**(1–2), 1–20 (2018)
9. Nagy, G.: Training a calligraphy style classifier on a non-representative training set. Electron. Imag. **2016**(17), 1–8 (2017)
10. Jiulong, Z., Luming, G., Su, Y., Sun, X., Li, X.: Detecting Chinese calligraphy style consistency by deep learning and one-class SVM. In: 2017 2nd International Conference Image, Vis. Computer ICIVC 2017, pp. 83–86 (2017)
11. Ye, Q., Doermann, D.: Text detection and recognition in imagery: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **37**(7), 1480–1500 (2015)
12. Rabi, M., Amrouch, M., Mahani, Z.: A survey of contextual handwritten recognition systems based HMMs for cursive arabic and latin script. Int. J. Comput. Appl. **160**(2), 31–37 (2017)
13. Azmi, M.S., Omar, K., Nasrudin, M.F., Wan Mohd Ghazali, K., Abdullah, A.: Arabic calligraphy identification for digital jawi paleography using triangle blocks. In: Proceedings 2011 International Conference Electronic Engineering Informatics, ICEEI 2011, July, pp. 1–5 (2011)
14. Bataineh, B., Abdullah, S.N.H.S., Omar, K.: Arabic calligraphy recognition based on binarization methods and degraded images. In: Proceedings 2011 International Conference Pattern Analysis Intelligence Robotics. ICPAIR 2011, vol. 1, pp. 65–70 (2011)
15. Azmi, M.S., Omar, K., Nasrudin, M.F., Muda, A.K., Abdullah, A.: Arabic calligraphy classification using triangle model for Digital Jawi Paleography analysis. In: Proceedings 2011 11th International Conference Hybrid Intelligence System, HIS 2011, pp. 704–708 (2011)
16. AlSalamah, S., King, R.: Towards the machine reading of arabic calligraphy: a letters dataset and corresponding corpus of text. In: 2nd IEEE Workshop on Arabic and Derived Script Analysis and Recognition, ASAR 2018, pp. 19–23 (2018)
17. Bin, P.G., Bin Talal, M.: Free Islamic Calligraphy (2012). https://freeislamiccalligraphy.com/

18. Makandar, A., Halalli, B.: Image enhancement techniques using highpass and lowpass filters. Int. J. Comput. Appl. **109**(14), 21–27 (2015)
19. Chen, Q., Sen Sun, Q., Ann Heng, P., Shen Xia, D.: A double-threshold image binarization method based on edge detector. Pattern Recognit. **41**(4), 1254–1267 (2008)
20. Gui, Y., Bai, X., Li, Z., Yuan, Y.: Color image segmentation using mean shift and improved spectral clustering. EURASIP J. Adv. Signal Process. **2012**(December), 5–7 (2012)
21. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: object localization by efficient subwindow search. In: 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)
22. Bay, H., Ess, A., Tuytelaars, T., Vangool, L.: Speeded-Up robust features (SURF) (Cited by: 2272). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
23. Dalal, N., Triggs, B., Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection to cite this version: histograms of oriented gradients for human detection. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. **1**, 886–893 (2005)
24. Haroon, M.: Comparative analysis of stemming algorithms for web text mining. Int. J. Mod. Educ. Comput. Sci. **10**(9), 20–25 (2018)
25. Nguyen, N.T.H., Soto, A.J., Kontonatsios, G., Batista-Navarro, R., Ananiadou, S.: Constructing a biodiversity terminological inventory. PLoS ONE **12**(4), 1–23 (2017)
26. Zhang, J., Guo, M., Fan, J.: A novel CNN structure for fine-grained classification of Chinese calligraphy styles. Int. J. Doc. Anal. Recognit. **22**(2), 177–188 (2019)