A Path Signature Approach to Online Arabic Handwriting Recognition

Daniel Wilson-Nunn, Terry Lyons, Anastasia Papavasiliou, Hao Ni

January 2018

Abstract

The Arabic script is one that has many properties that come together and result in what is commonly cited as one of the most beautiful scripts. Used by over 400 million people worldwide and with a history spanning over 1800 years, the Arabic script remains one of the most important languages in the world. Using tools from the theory of rough paths, combined with state of the art techniques from deep learning, we develop a recognition methodology for Arabic handwriting. Preliminary results using online Arabic handwritten characters show that the methodology developed can result in a significant decrease in error rate.

1 Introduction

Handwriting recognition methods are of two distinct types; online and offline. Online handwriting recognition deals with data recorded "in time", i.e. data that is represented as a function of time. Offline handwriting recognition, on the other hand, deals with recognition of data that is in image format. Online handwriting recognition deals with the spatio-temporal resolution of the input whereas offline handwriting recognition deals with the spatio-luminance of an image [1]. Handwriting recognition has a number of important application areas, from converting handwriting to text on a tablet or touch screen to signature verification for bank fraud to digitisation of ancient manuscripts for historical study [2].

Recent advances in computing and deep learning have resulted in a large amount of interest in the area of hand- writing recognition, with over 17,000 results for "handwriting recognition" appearing on Google Scholar since 2013. The popularity of the field has lead to significant developments and advancements in recent years.

1.1 Arabic Script

Many different languages have a script based upon the Arabic alphabet, these include Pashto, Urdu, Persian, Jawi, Uyghur, Kurdish as well as a number in addition to these. Owing to the vast usage of the Arabic script, the ability to recognise Arabic handwriting is important and has applications relating to millions of people. The Arabic script is one with a number of challenges that add to the complexity of handwriting recognition:

1. *Cursive Nature*: The Arabic script comprises of 28 standard letter characters (with additions in regional variants). Arabic is always written cursive, regardless of whether it is handwritten or typed.
2. *Changing letter shapes*: Each of the 28 characters in the Arabic script can take a number of different shapes. This is a direct result of the cursive nature of the script and adds to the complexity of the script. Other cursive scripts such as Latin or Bengali often have a fixed letter shapes with small variations, however Arabic characters may change drastically depending on their location in the word.
3. *Joining and non-joining characters*: Of the 28 standard characters, 6 are "non-joining" which means that although the script is cursive, they do not join to the following letter. This will result in a small space within a word without the word actually ending.
4. *Delayed strokes*: Arabic is heavily reliant on delayed strokes, specifically dots, that are added to letters to distinguish them from each other. Whilst this is less of a concern in offline recognition, for online recognition, this adds significant difficult.
5. *Ligatures*: In addition to changing letter shapes, there are a number of combinations of letters that form ligatures. One ligature that always occurs, as opposed to being part of different handwriting styles is ل followed by ا, instead of becoming لـا, it becomes لا.

As an example, to illustrate the above points, the two words forming the phrase بيت الملك (*bait al malik* - the king's house) contain the letters ب، ي، ت، ا، ل، م، ل، ك. Notice the following:

1. Within each word, the majority of letters are joined to each other
2. The letter ي is very different in its isolated form to the shape it takes in the middle of the word بيت
3. In the word الملك, the letter ا is not connected to the letter following it. This is because it is one of the 6 "non-joining" letters
4. In the word بيت, the delayed strokes, in the form of dots, are used to distinguish the letters from the word تيب for example which has the same underlying shape but different dots
5. When the letters ل and م are joined, followed by another letter, they form the ligature لمـ as opposed to لم. Notice that the ligature formed by ل and م is common however not always used.

In this paper, a methodology for transforming online handwritten character data for use in classification is proposed. This work has concentrated on online Arabic character recognition, the methodology proposed may also have application for online phrase recognition. In part III. Methodology of this paper, the methodology developed will be outlined and in part IV. Results, details of the performance of this methodology are presented.

2 Online Handwriting & Data Used

Online handwritten characters are sequences of (x, y) coordinates, possibly with multiple sequences per character to indicate multiple strokes in a single character. The general format of these characters is shown in (1).

$$X_i = \{\underbrace{\overbrace{\left[(x,y)_1,(x,y)_2,\dots,(x,y)_{\ell_1}\right]_1}^{\substack{\text{Arbitrary number of coordinates per stroke}\\ \uparrow}},\dots,\left[(x,y)_1,(x,y)_2,\dots,(x,y)_{\ell_N}\right]_N}_{\substack{\downarrow\\ \text{Arbitrary number of strokes per character}}},Y_i\}, \tag{1}$$

where Y_i is the class of the character X_i.

The data used in this section comes from the Online KHATT dataset with individual characters segmented. This data has been produced and worked on by al Hilali and Mahmoud [3].

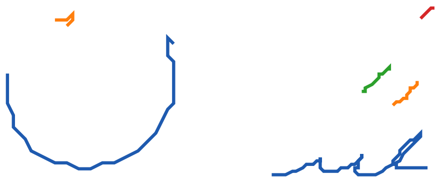

Figure 1: Online handwritten Arabic characters (ن and شـ)

The dataset contains 22,795 observations, falling into 57 different classes. The dataset is split into training and testing sets as shown in Table 1.

The remaining data is kept for validation. Of the 50 classes in the testing data, all are present in the training data.

Set	Size	No. Classes
Training	11397	52
Testing	6833	50

Table 1: Breakdown of splitting the dataset into training and testing set

2.1 Background

Online handwriting recognition has been an area of study for a number of years. There are many different methodologies that have been proposed to classify handwritten characters. Historically, the main classification tool was Hidden Markov Models (HMM). In recent years, the abilities of Neural Net- works have been significantly improved as a result of increases in computation abilities.

1. *Hidden Markov Models*: An HMM is a tool that is used to express probability distributions on sequences of observations of a time series. These discrete observations of the time series form a discrete stochastic process Y_t, where the structure of Y_t is of any form (real valued, integer, letters etc.) with the restriction that it is possible to define a probability distribution on the process.
2. *Neural Networks*: Neural networks are an attempt to mimic the way in which the brain works. Through multiplication of input data by a large number of weights at nodes, followed by an optimisation of the weights so as to achieve the best possible outcome, neural networks have achieved significant prominence in the field of classification owing to their abilities.

3 Methodology

The methodology presented below transforms the raw character data (assumed to lie in $[-1,1]^2$) to a set of feature variables that can be used for classification with a tool such as LSTM.

3.1 Fixing dimensionality

Each character contains a number of strokes and each stroke contains a number of coordinates (x, y). The first step is to change the character from a list of strokes to a single list of coordinates. To do this, a third dimension is added, and a number of “jump points” are also added. For ease of notation, let $(x, y)_i^k$ be the i-th coordinate of stroke k. The third dimension, P, is added through the following mechanism:

$$(x, y)_i^k \mapsto ((x, y)_i^k, P) \equiv (x, y, P)_i^k,$$

where

$$P = 2 \times (k - 1)\,,$$

where k is the number of the stroke that contains the coordinate. For example, the coordinate $(0.1, 0.5)_2^2$ (i.e. the second coordinate in stroke 2) would become $((0.1, 0.5)_2^2, 2) \equiv (0.1, 0.5, 2)_2^2$. The data is now of the form:

$$\begin{aligned} X_i = \{&((x,y)_1^1, 0), \dots, ((x,y)_{\ell_1}^1, 0), \\ &((x,y)_1^2, 2), \dots, ((x,y)_{\ell_2}^2, 2), \\ &\dots \\ &((x,y)_1^N, 2N-2)), \dots, ((x,y)_{\ell_N}^N, 2N-2)\}. \end{aligned} \tag{2}$$

The second step to this stage is to add coordinates to indicate where the pen leaves the paper (tablet). These coordinates are added between coordinates where a new stroke begins and have the exact same x and y coordinate values as the end point of one stroke and beginning point of the next. The P value of these two coordinates that are added is equal to the odd integer that lies between the two even integers of the existing coordinates. For example, the data in (2) would become

$$\begin{aligned} &X_i = \{((x,y)_1^1, 0), \dots, ((x,y)_{\ell_1}^1, 0), \\ \text{new} \quad \rightarrow \quad &((x,y)_{\ell_1}^1, 1), ((x,y)_1^2, 1), \\ &((x,y)_1^2, 2), \dots, ((x,y)_{\ell_2}^2, 2), \\ &\dots \\ \text{new} \quad \rightarrow \quad &((x,y)_{\ell_{N-1}}^{N-1}, 2N-3), ((x,y)_1^N, 2N-3), \\ &((x,y)_1^N, 2N-2)), \dots, ((x,y)_{\ell_N}^N, 2N-2)\}. \end{aligned} \tag{3}$$

3.2 Signature of the Character

After the data has been converted to a single series of coordinates in $\mathbb{R}^3$, the data extraction method in preparation for the classifier is to be carried out. The method that will be used in this work is know as dyadic signatures., developed based on the rough path signature, invented by Terry Lyons [4].

Definition 1. *Given a path $X : [S,T] \to \mathbb{R}^d$, the signature of the path over an interval $[s,t]$ is defined as*

$$S_{[s,t]}(X) = \left(1, X^1_{[s,t]}, X^2_{[s,t]}, \dots, X^n_{[s,t]}, \dots\right),$$

where $X^n_{[s,t]}$ is defined as

$$X^n_{[s,t]} := \int\limits_{\substack{u_1 \leq \cdots \leq u_n \\ u_1, \dots, u_n \in [s,t]}} \mathrm{d}X_{u_1} \otimes \cdots \otimes \mathrm{d}X_{u_n}.$$

Note that it is necessary to only take the signature up to a certain level, i.e.

$$S^{(N)}_{[s,t]}(X) = \left(1, X^1_{[s,t]}, X^2_{[s,t]}, \dots, X^N_{[s,t]}\right).$$

The level N signature of a path on $\mathbb{R}^d$ lies in the space $T^{(N)}\left(\left(\mathbb{R}^d\right)\right)$. This space provides well defined logarithm and exponential functions, and the logarithm of $S^{(N)}_{[s,t]}(X)$ is an important tool, known as the log signature.

$$\log\left(S^{(N)}_{[s,t]}(X)\right) = s^{(N)}_{[s,t]}(X).$$

3.3 Using the Signature

The first step to using the signature to encode details of the characters is to take a linear interpolation - at constant speed - of the sequence of points in 3D that were produced in the first stage. Second is to consider this as a linear path in 3D and to compute the signature of each character (up to a fixed level, say M). The resulting data is used as the input for a classifier.

In order to use *dyadic* signatures, instead of calculating the signature of the whole character, the character is split into 2^k sections of equal length and the signature of each of these sections is calculated and used as input for the classifier.

4 Results

Using a Long Short Term Memory (LSTM) network, with three hidden layers, each containing 50 nodes, the recognition is carried out. The model is trained using TensorFlow on an NVidia GTX 1080Ti graphics card and the results for different signature levels and number of dyadic intervals are obtained as follows.

Run	**Signature Level**	**Dyadic Level**	**Recognition Rate (%)**
1	2	3	90.44%
2	5	5	92.57%
3	2	5	91.26%
4	5	2	90.57%
5	10	2	90.56%
6	10	4	91.38%

Table 2: Results of LSTM networks trained using various parameters

The results show that we obtain a recognition rate of over 92.5% when using signature level 5 and 32 (2^5) dyadic intervals of the character. This rate is significantly improved on the 82% achieved by Hilali and Mahmoud in their original paper, thus showing the capabilities of the method on this data.

5 Conclusion

It has been demonstrated that the path signature approach to online Arabic character recognition when combined with LSTM, has the ability to perform very well. A number of opportunities present themselves as a result of this methodology, including the ability for very fast recognition of new characters, owing to the fact that computation of the signature is very fast and capable of being done on smartphone processors.

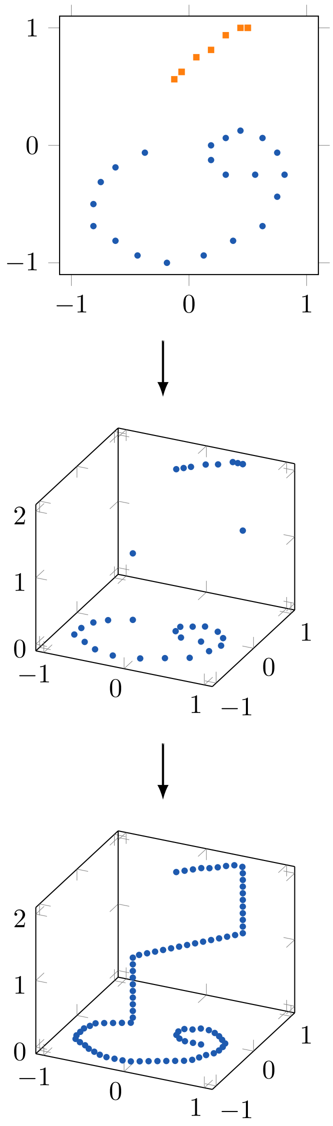

Figure 2: Transformation from original character to linearly interpolated character with “pen” dimension added (character ق)

Acknowledgement

I wish to thank my three supervisors, as well as the Alan Turing Institute for the funding provided to carry out this research. Additionally thanks to Sabri Mahmoud for providing the data used. Furthermore I wish to thank Prof Najoua Essoukri Ben Amara for her mentoring as part of the ICDAR 2017 Doctoral Consortium.

References

[1] R. Plamondon and S. N. Srihari, “Online and off-line handwriting recognition: a comprehensive survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, Jan. 2000.

[2] A. Priya, S. Mishra, S. Raj, S. Mandal, and S. Datta, “Online and offline character recognition: A survey,” in *2016 International Conference on Communication and Signal Processing (ICCSP)*, Apr. 2016, pp. 0967–0970.

[3] B. M. Al-Helali and S. A. Mahmoud, “A Statistical Framework for Online Arabic Character Recognition,” *Cybernetics and Systems*, vol. 47, no. 6, pp. 478–498, Aug. 2016. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01969722.2016.1206768

[4] I. Chevyrev and A. Kormilitzin, “A Primer on the Signature Method in Machine Learning,” *arXiv:1603.03788 [cs, stat]*, Mar. 2016, arXiv: 1603.03788. [Online]. Available: http://arxiv.org/abs/1603.03788

[5] B. M. Al-Helali and S. A. Mahmoud, “Arabic Online Handwriting Recognition (AOHR): A Survey,” *ACM Comput. Surv.*, vol. 50, no. 3, pp. 33:1–33:35, Jun. 2017. [Online]. Available: http://doi.acm.org/10.1145/3060620

[6] J. Reizenstein, “Calculation of Iterated-Integral Signatures and Log Signatures,” *arXiv:1712.02757 [math]*, Dec. 2017, arXiv: 1712.02757. [Online]. Available: http://arxiv.org/abs/1712.02757

[7] F. Biadsy, J. El-Sana, and N. Habash, “Online Arabic Handwriting Recognition Using Hidden Markov Models.” Suvisoft, Oct. 2006. [Online]. Available: https://hal.inria.fr/inria-00108306/document

[8] H. E. Abed, V. Märgner, M. Kherallah, and A. M. Alimi, “ICDAR 2009 Online Arabic Handwriting Recognition Competition,” in *2009 10th International Conference on Document Analysis and Recognition*, Jul. 2009, pp. 1388–1392.

[9] B. Alsallakh and H. Safadi, “AraPen: An Arabic Online Handwriting Recognition System,” in *2006 2nd International Conference on Information Communication Technologies*, vol. 1, 2006, pp. 1844–1849.

[10] I. Abdelaziz, S. Abdou, and H. Al-Barhamtoshy, “A large vocabulary system for Arabic online handwriting recognition,” *Pattern Analysis and Applications*, vol. 19, no. 4, pp. 1129–1141, Nov. 2016. [Online]. Available: https://link.springer.com/article/10.1007/s10044-015-0526-7

[11] T. Lyons, C. Ladroue, S. Buckley, G. Gyurko, C. Litterer, A. Janssen, and C. Xu, “Computational Rough Paths.” [Online]. Available: http://coropa.sourceforge.net/

[12] M. Kherallah, L. Haddad, A. M. Alimi, and A. Mitiche, “On-line handwritten digit recognition based on trajectory and velocity modeling,” *Pattern Recognition Letters*, vol. 29, no. 5, pp. 580–594, Apr. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865507003662

[13] K. Assaleh, T. Shanableh, and H. Hajjaj, “Recognition of handwritten Arabic alphabet via hand motion tracking,” *Journal of the Franklin Institute*, vol. 346, no. 2, pp. 175–189, Mar. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0016003208001038

[14] N. Tagougui, H. Boubaker, M. Kherallah, and A. M. ALIMI, “A Hybrid NN/HMM Modeling Technique for Online Arabic Handwriting Recognition,” *arXiv:1401.0486 [cs]*, Jan. 2014, arXiv: 1401.0486. [Online]. Available: http://arxiv.org/abs/1401.0486

[15] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Taylor & Francis, Jan. 1984.

[16] N. Tagougui, M. Kherallah, and A. M. Alimi, “Online Arabic handwriting recognition: a survey,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 16, no. 3, pp. 209–226, Sep. 2013. [Online]. Available: https://link.springer.com/article/10.1007/s10032-012-0186-8

[17] “Ink Markup Language (InkML).” [Online]. Available: https://www.w3.org/TR/InkML/

[18] W. Yang, L. Jin, Z. Xie, and Z. Feng, “Improved Deep Convolutional Neural Network For Online Handwritten Chinese Character Recognition using Domain-Specific Knowledge,” *arXiv:1505.07675 [cs]*, May 2015, arXiv: 1505.07675. [Online]. Available: http://arxiv.org/abs/1505.07675

[19] Z. Xie, Z. Sun, L. Jin, Z. Feng, and S. Zhang, “Fully Convolutional Recurrent Network for Handwritten Chinese Text Recognition,” *arXiv:1604.04953 [cs]*, Apr. 2016, arXiv: 1604.04953. [Online]. Available: http://arxiv.org/abs/1604.04953

[20] T. Lyons, “Rough paths, Signatures and the modelling of functions on streams,” *arXiv:1405.4537 [math, q-fin, stat]*, May 2014, arXiv: 1405.4537. [Online]. Available: http://arxiv.org/abs/1405.4537

[21] B. Graham, “Sparse arrays of signatures for online character recognition,” *arXiv:1308.0371 [cs]*, Aug. 2013, arXiv: 1308.0371. [Online]. Available: http://arxiv.org/abs/1308.0371