11-2017

Isolated Character Forms from Dated Syriac Manuscripts

Nicholas Howe
Smith College, nhowe@smith.edu

Minyue Dai
Smith College

Michael Penn
Stanford University

Isolated Character Forms from Dated Syriac Manuscripts

Nicholas R. Howe
Smith College
Northampton, Massachusetts
nhowe@smith.edu

Minyue Dai
Smith College
Northampton, Massachusetts
mdai@smith.edu

Michael Penn
Stanford University
Stanford, California
mppenn@stanford.edu

ABSTRACT

This paper describes a set of hand-isolated character samples selected from securely dated manuscripts written in Syriac between 300 and 1300 C.E., which are being made available for research purposes. The collection can be used for a number of applications, including ground truth for character segmentation and form analysis for paleographical dating. Several applications based upon convolutional neural networks demonstrate the possibilities of the data set.

CCS CONCEPTS

• **Applied computing** → *Digital libraries and archives*; Arts and humanities;

KEYWORDS

Syriac, paleography, historical manuscripts, handwriting style

ACM Reference Format:
Nicholas R. Howe, Minyue Dai, and Michael Penn. 2017. Isolated Character Forms from Dated Syriac Manuscripts. In *Proceedings of The 4th International Workshop on Historical Document Imaging and Processing, Kyoto, Japan, November 10–11, 2017 (HIP2017)*, 6 pages.
https://doi.org/10.1145/3151509.3151513

1 INTRODUCTION

Written Syriac is a dialect of Aramaic that appeared in the first century C.E. and remains in use today in some areas of the Middle East. During the first millenium C.E. it became a major liturgical language for Christians, alongside Latin and Greek, and was adopted by the Syriac Orthodox Church and the Church of the East. As a result of this widespread use and favorable environmental conditions, tens of thousands of historic manuscripts written in Syriac script survive to this day and may be found in the collections of monasteries and libraries across the world.

Despite this significance, Syriac has received much less attention to date from researchers than Latin or Greek, and the number of scholars literate in the language today remains small. Modern scholarship would benefit greatly from the development of automated techniques for processing the language, so that existing manuscript collections could be systematically studied.

To advance study of the language and automated methods for processing it, this paper describes a collection of Syriac character images that have been binarized and segmented by hand, a portion of which is being made publicly available for research purposes.[1] This work extends a previous paper that introduced a similar set of Syriac characters with bounding boxes that were not yet publicly available [8]. The new data set includes more annotations plus full character segmentations in addition to bounding boxes, and is now ready for partial public release.

In addition to describing the data set, this paper looks at several applications enabled by it, including manuscript dating and learned character segmentations. Section 2 describes the data set itself and its creation. Section 3 describes the related experiments. Finally, Section 4 concludes with a short summary.

Table 1: Manuscripts By Source

Source	# Dated	# Undated
British Library	113	2
Vatican Library	18	42
Other	25	24
Total	156	68

2 SYRIAC CHARACTER DATA SET

Potential uses for isolated images include character recognition, segmentation, and manuscript dating. To facilitate the latter, source documents are chosen that have been securely dated through scribal notes or similar evidence. There exist roughly 200 manuscripts with secure dates prior to 1200 C.E. [1]. Of these, copies of 156 have been secured for analysis, and 131 are included in the released data set. An additional 68 undated manuscripts have also been secured, of which 44 are in the released set. Table 1 summarizes the collected manuscripts by source, and Figure 1 visualizes their dates on a timeline. Table 2 gives the number of samples for each letter category.

Character samples were extracted by hand from each document in a multi-step procedure. First, bounding boxes were identified for each Syriac character in each document, as described in prior work [8]. Known common variants of each letter were identified separately at this stage, where present. Figure 2 shows characters of the Syriac alphabet, including an example of each of the different variant categories that were identified. Not all forms are present in

HIP2017, November 10–11, 2017, Kyoto, Japan

ACM ISBN 978-1-4503-5390-8/17/11…$15.00
https://doi.org/10.1145/3151509.3151513

[1] The selected documents are owned by a range of institutions, under usage terms that do not always allow public release. However, the two institutions holding the largest set of documents have graciously agreed to share character images from their collections for research purposes. We thank the British and Vatican Libraries for their generosity.

Figure 1: Timeline showing distribution of dates for the manuscripts included in the study.

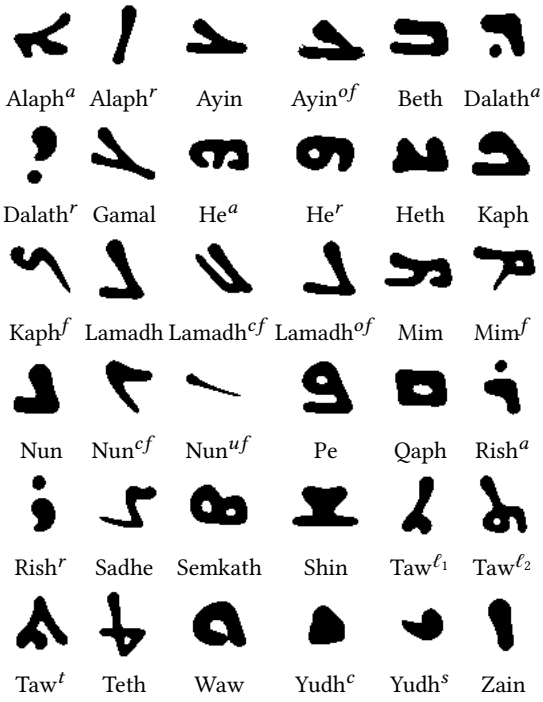

Figure 2: Samples of each of the character categories identified in this study. Although Syriac has only 22 letters, there are 36 different forms when letter variants are included. Annotations: a = angular, r = round, f = final, c = connected, u = unconnected, o = open, ℓ_1 = L-shaped, ℓ_2 = looped, t = triangular, s = stand-alone. This work categorizes style at the level of individual character forms rather than entire manuscripts, because the dataset shows that many manuscripts mix forms traditionally characterized as solely Estrangela or Serto.

each manuscript; for example, usually just one of the rounded or angular form is used.

Following bounding box identification, the document images were binarized using Howe's algorithm [9] and cropped versions of each letter were extracted. These cropped images, while containing the entire target letter centered within the frame, often include bits of other characters around the edges, and also may be attached to them via connecting strokes. Prior work has examined automatic methods for removing the extraneous markings, with reasonable success [8]. However, to avoid introducing inadvertent bias to any further processing that might be performed, a ground truth segmentation is required. This was produced by human labor,

Figure 3: Stages in the creation of the character image data set. First row ©British Library Board, BL. Add. 12,145, f. 3a: original image in the vicinity of the specified bounding box (rectified image). Second row: binarized version. Third row: edited version with extraneous marks removed.

again under the supervision of a language expert. The workers used image processing software to delete extraneous marks and sever connecting strokes at an appropriate point, taking around three hours per manuscript to produce four to six examples of each character. Because isolating characters necessarily eliminates the ligatures between them, this work focuses on the morphology of the letters alone and leaves for future research the analysis of information contained in the connecting strokes. Figure 3 shows stages of the editing process for several sample letters.

The raw samples vary in size, aspect ratio, and resolution according to the source document and letter characteristics. Many of the experiments described in the next section require size consistency in the input images. The median sample dimensions are around 60 in each dimension, so a rectified 60×60 sample set has also been created where needed. The rectified samples have all been scaled and resampled with interpolation to consistent size. In all cases where the original bounding box was not square, the longer dimension is scaled to 60 pixels and the shorter dimension is expanded by adding equal margins on either side as necessary to make a square. The original image is then sampled using the new square bounding box to get the rectified character sample, and likewise with the binary image.

The British Library and Vatican Library have graciously granted permission to release the set of segmented character images from their manuscripts for use in research. This subset of data will be available by request via the first author's web page, along with details of character bounding box coordinates and other related data.

3 EXPERIMENTS

This section presents the results of several experiments using the new data set, all based upon trained convolutional neural networks (CNN). The applications chosen may be of interest in their own right, but are also intended as possible baselines for future work. The subsections below describe three applications: character recognition, character segmentation, and style detection for manuscript dating. Since all rely on the same network architecture, the discussion begins with its description.

3.1 Network Architecture

All three applications rely on a convolutional neural architecture arranged as a conditional generative adversarial network (GAN) [3, 13]. Such networks consist of two main subcomponents, called the discriminator and the generator respectively, which are trained simultaneously. Training takes the form of an adversarial game: the goal of the generator is to produce fake images that are as realistic as possible in order to fool the discriminator; meanwhile the discriminator seeks to distinguish the fake inputs from real ones. The same components can also be rearranged into an autoencoding configuration, where the discriminator encodes an image into a condensed representation, and the generator decodes it to reproduce the original image.

The architectures of the two units mirror each other, as shown in Figure 4. The discriminator/encoder architecture consists of three convolutional layers with 2×2 pooling layers in between. The convolutional layers use a 5×5 kernel with a stride of 2, and depth increases as shown in later layers. Following these are two fully-connected layers, connected to an output that is either a binary discrimination result (for the discriminator) or a z vector of $k = 100$ dimensions (for the encoder). The character label is made available at all levels as an additional input; because the network is implemented using the TensorFlow architecture this is implemented by augmenting the data with extra nodes or layers activated in a one-hot configuration.

The generator/decoder takes a z vector of similar size (generated from a random normal distribution when creating fake images) back through two fully connected layers into a low-resolution image layer, which then passes through unpooling and deconvolution layers that mirror the structure on the other side. Although they share a similar architecture, the connection weights of the encoder and decoder are not coupled in any way.

In practice, effective training requires several phases. Prior to the adversarial phase, the discriminator is first trained at a character recognition task using the available labeled data. This gives it some basic structure for encoding character images, culminating in a vector of activations z. During adversarial training, the discriminator is given either a real training image or a fake image produced by the generator, and must tell the two conditions apart. The generator meanwhile learns to reproduce a training image from the z encoding produced in the discriminator, or updates itself to better fool the discriminator when one of its own randomly generated images is chosen.

For reproducibility, specific training details follow. The clean character images are all scaled and translated using the procedure known as congealing [10] to minimize their common entropy across each character class. All network weights are initialized from normal distribution centered to 0 with standard deviation as 0.02. The leak of the Leaky ReLU [12] is set to 0.2. In the optimization process, the Adam optimizer [11] is used to accelerate the training. The learning rate is set to 0.0002 to ensure the convergence of the training result. If z represents the randomly generated latent representation and ℓ represents the class label, let $G(z, \ell)$ represent the output image from the generator and $D(I, \ell)$ the output probability from the discriminator that the input I is a real image. The training process's goal is to maximize the objectives Φ_G for the generator

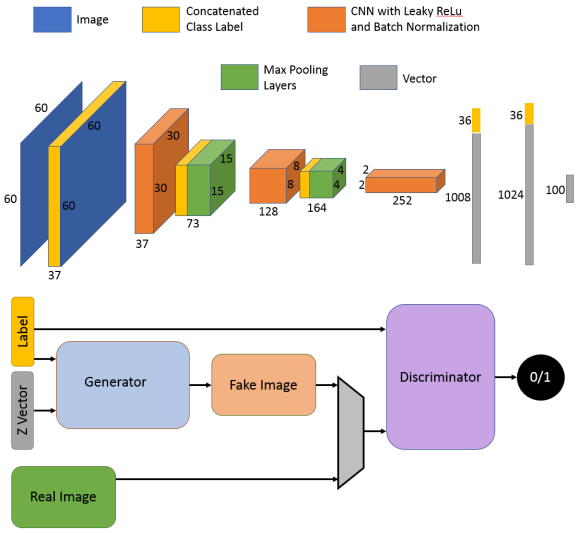

Figure 4: Architecture of the network components. Top shows layers in the encoder unit, which is identical to the discriminator except for the output layer. Decoder unit architecture is inverse of the encoder. Bottom shows the generative adversarial network architecture with generator and discriminator units in context.

and Φ_D for the discriminator, given below.

$$\Phi_G = -\log(1 - D(G(z, \ell), \ell)) \tag{1}$$

$$\Phi_D = \log(D(I, \ell)) + \log(1 - D(G(z, \ell), \ell)) \tag{2}$$

In practice, G cannot learn well until D is sufficiently trained to provide a meaningful gradient. Therefore D trains alone during a pre-training phase, using real images and randomly generated images from untrained G. Once D achieves some level of accuracy, regular training begins on both networks in parallel.

3.2 Character Classification

As mentioned above, the same basic network components can be reconfigured and applied to several other relevant tasks. To perform character classification, a new network is trained from scratch using just the encoder module, minus the class label inputs. Output is 36 nodes in one-hot configuration.

We train and test character classifiers using two different types of input. The first is the cropped raw page image (equivalent to the first row in Figure 3), either RGB or grayscale depending on the source. The second is the hand-segmented, scaled and centered data (visible in the third row). Note that the first task is more difficult due to the extra noise, but it is not quite equivalent to character recognition "in the wild" because the window has been scaled and resampled with knowledge of the size of the actual character. Syriac letters can vary dramatically in size; a square window containing an entire *nun* can also hold several other complete or nearly-complete additional letters due to the width of the target. The second task using hand-extracted letters is even more artificial, but potentially serves as a Syriac analog to the well-known MNIST benchmark. (It

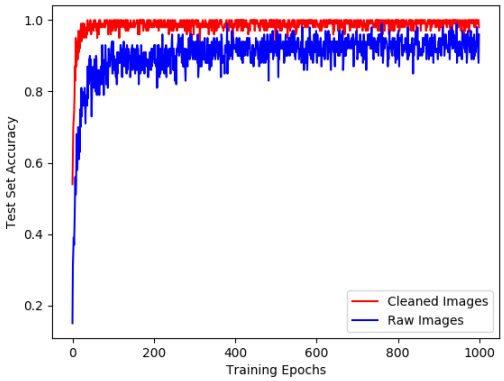

Figure 5: Test set accuracy vs. training duration

is actually more difficult than MNIST, having 36 classes and larger images.)

For each condition, 10000 randomly selected images form the test set, and the remaining 50000+ images are used for training. After 1000 training epochs, the test set accuracy averages 93.2% on the raw images and 99.2% on clean images. Figure 5 shows progress on test set accuracy as training progresses.

3.3 Image Cleaning & Character Segmentation

For the character segmentation/denoising application, we use the original autoencoder network architecture and train with examples of the desired task as input and output. Given images from the first row of Figure 3, the system must produce images that look like the third row. Figure 6 shows some examples. This task uses the same train/test split as for character classification. The network produces images with fractional values at most pixels; these are thresholded to produce a final binary result. We test two similar network architectures, one with access to the target character class label and one without access to this information.

The results can be summarized numerically by computing an F-measure, using the hand labeled ground truth to determine the number of pixels with true positive, false positive, and false negative weights. Figure 7 shows the mean F-measure evaluated on the test set as training progresses for both networks. The final mean F-measure values are 82% and 79%, showing that the character label input makes a slight difference, but not much. Although this is a new data set, prior work by Howe et al. using a different method on similar data reported qualitative results that offer some point of comparison [8]. That paper reports significant errors in about 10% of test images, and minor errors in around 40%.

Figure 8 shows some representative errors made by the network. Although errors this large occur in a small minority of cases, they show areas where improvement is clearly possible.

3.4 Style Comparison & Date Estimation

The encoder network described above encodes any character image I into a lower-dimension vector z. The generative aspect of the system ensures that z sits in a space Z that embeds all of the style

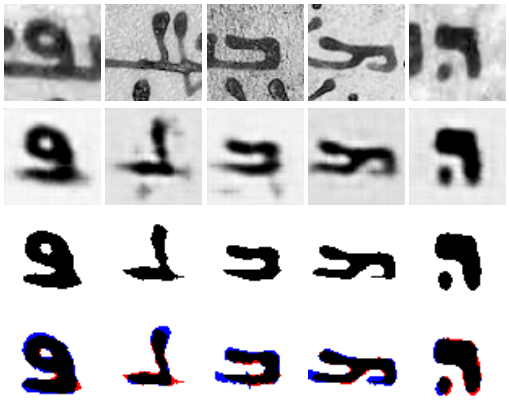

Figure 6: Steps in the image segmentation/denoising process. Top row ©British Library Board, BL. Add. 12,145, f. 3a. shows the input image (false color), with output after training in the second row. Third row shows the thresholded result, and fourth row superimposes the target.

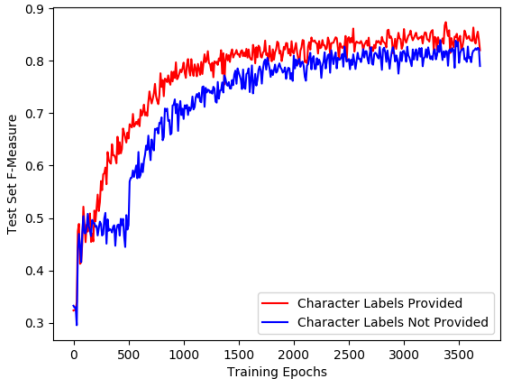

Figure 7: F-measure evaluated on the test set vs. training duration for two network architectures.

and shape variation present in the training data, and potentially interpolates to additional unseen combinations. The z vector of a character sample thus carries useful information about the writing style, which can be used for manuscript dating. We begin with some preliminary experiments to verify this hypothesis. If true, then characters of the same type from the same manuscript should cluster closely together. One way to verify this is by running a retrieval experiment, using each character as a query and the remaining samples of that character from the same manuscript as targets. All experiments in this section use the clean hand-extracted images.

Table 2 summarizes the results of same-manuscript character retrieval run over all samples and manuscripts. The mean average precision over all the trials is 37%. This indicates a moderate degree of consistency in the z vectors.

Because several sets of manuscripts are known to have been written by the same scribe, it is also possible to test whether characters

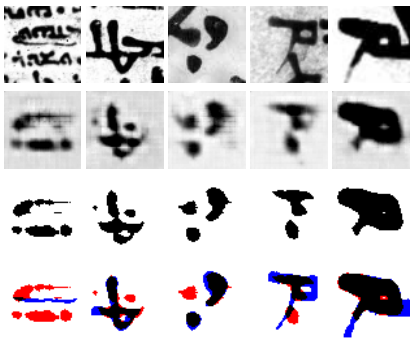

Figure 8: Some representative failure modes. In the first two examples the input image is quite cluttered, with more noise than signal. In the next two the network fails to completely erase a large distractor that is close to the target letter. In the final example, a portion of the target letter that is near the edge of the window is discarded as noise. Top row ©British Library Board, BL. Add. 12,145, f. 3a.

from different manuscripts in the same hand will also cluster. Five such groupings are known to exist in the collection, ranging from two to four manuscripts in size. Table 3 summarizes the results of a recall-precision test on the full set of characters from each of these groups. The mean average precision over all the trials is 19.3%, which is lower than the result within a single manuscript, but still indicates some clustering. Of course, different manuscripts have been subject to varying storage conditions over the centuries and may have been produced at different points in the scribe's lifetime. For example, 30 years separate one pair of manuscripts, and another set spans 20 years. In both of these cases there is evidence of a change in scribal style, including the use of different letter forms.

The successful results above naturally raise the question of whether the z representation can be used to accurately date manuscripts. Hypothetically, manuscripts written in the same style of specific letters are more likely to have been produced under similar circumstances and hence nearby in time. The letters from such a cluster should all map to small localized regions in Z, and thus analysis of z vectors should provide useful clues for dating.

Prior work in manuscript dating has often begun by clustering characters [4–7, 14, 15]. Unfortunately, by design the Z space is inimical to clustering in its construction: the character style manifold maps densely onto Z without clear boundaries to provide guidance. Furthermore, popular clustering algorithms such as Gaussian mixture models behave quite differently in 100 dimensions than in two or three, and much more data is needed. For this reason the experiment herein adopts a different, non-parametric approach that does not require any automatic clustering.

As in Howe et al. [7] we assess the relevance of document A for dating document B by measuring the likelihood that each of their corresponding letter sample populations were drawn from the same distribution. We adopt a test from nonparametric statistics [2] for this purpose: given two sets of z vectors, compute the centroids and the distance between them. Next, perform a form of Monte Carlo sampling to assess the likelihood that they came from the same distribution: shuffle vectors between sets to create two new sets of the same size as the original, and compute the distance between their centroids. The one-tailed p-value for the original configuration is equal to the fraction of random shuffles with larger centroid distance than the original.

Table 2: Mean Average Precision for Same-Character Recall Within Manuscripts

Character	MAP	# Samples
Alaph (Angular)	46.2	1394
Alaph (Round)	59.9	475
Ayin	35.4	1394
Ayin (Final, open)	100.0	2
Beth	36.8	2806
Dalath (Angular)	34.9	1107
Dalath (Round)	43.1	1013
Gamal	38.5	2395
He (Angular)	41.4	1014
He (Round)	43.2	1035
Heth	31.1	2699
Kaph	39.5	1900
Kaph (Final)	41.7	1701
Lamadh	28.6	3220
Lamadh (Final, closed)	52.8	331
Lamadh (Final, open)	40.4	986
Mim	37.5	2399
Mim (Final)	36.7	1753
Nun	30.4	2217
Nun (Final, connected)	38.3	1694
Nun (Final, unconnected)	51.2	1721
Pe	30.6	2677
Qaph	31.3	2619
Rish (Angular)	41.4	1018
Rish (Round)	39.7	1050
Sadhe	34.2	1987
Semkath	33.4	2266
Shin	37.1	1480
Taw (L-shaped)	51.4	358
Taw (Looped)	37.0	1552
Taw (Triangular)	35.2	537
Teth	39.4	2288
Waw	27.5	4132
Yudh (Connected)	41.4	2105
Yudh (Stand-alone)	38.4	1626
Zain	48.3	1936

Table 3: Mean Average Precision for Same-Character Recall Across Manuscripts by the Same Scribe

Scribe	MAP	# Manuscripts
Saba of Ras'ain	23.7%	3
George	16.9%	2
Addai of Amid	25.0%	2
Yesya	15.1%	4
Samuel b. Cyriacus	16.0%	4

For two sets both drawn from a single distribution (the null hypothesis), this p-value will be uniformly distributed between 0 and 1. On the other hand if the sets are drawn from two different distributions (the alternative hypothesis), then smaller values of p become much more likely. To estimate a new distribution of p-values for the latter situation we again turn to simulation. By design, all components of z within a normal distribution of variance 1.0 correspond to plausible models of a character. However, the sample sets drawn from individual manuscripts tend to show much smaller variance, around 0.15 on average. The simulation procedure becomes thus: select two centroids from a standard normal distribution, and use these to generate individual samples from normal distributions of variance 0.15 located at the previously chosen centroids. Once sample sets have been chosen, compute the nonparametric p-value. The resulting distribution of p-values differs greatly from the situation under the null hypothesis: all the probability mass lies below a p-value of 1%. To compute the likelihood that two actual sample sets are drawn from the same distribution, simply compare the relative probability mass of the null and alternative hypotheses at the computed p-value of the two sets.

Qualitatively, the procedure outlined above can give results that seem too conservative, concluding letter sets to be different on the basis of seemingly small details. This is partly a consequence of modeling the alternative hypothesis using manuscript-scale clusters. Human intuition might accord more closely with larger clusters that encompass such minor differences. More important than subjective impressions is performance in application: how well can it date manuscripts?

Using the Bayesian framework described by Howe et al. [7] to compute a temporal profile for each manuscript, we identify the year at the 50^{th} percentile of the probability mass, Y_{P50}. The mean error of this date estimate for the 156 available dated manuscripts is 144 years, somewhat worse than the result given in prior work (116 years). However, this number hides a subset on which the method does much better: the median error is only 70 years, and the best quartile has an error below 25 years. By contrast, the method used in the prior work shows similar mean and median error, both over 100 years.

Investigation shows that nearly all of the most accurate date predictions come from identification of one similar manuscript written around the same time. In some of these cases both manuscripts are known to be the work of a single scribe, thus explaining the close similarity; in other cases the scribes are unknown. Curiously, the dating error does not seem to be well correlated with measured levels of document similarity. Figure 9 shows the error on individual documents plotted against the strength of that document's relationship to all others. Accurate dating occurs at all levels of certainty.

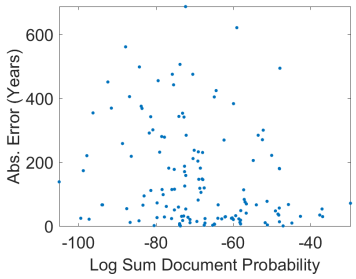

Figure 9: Absolute error of date predictions plotted against log sum document relatedness probabilities.

4 CONCLUSION

The primary contribution of this paper is the introduction of a new collection of highly curated character samples taken from securely dated manuscripts written in Syriac between roughly 400 and 1200 CE. Comprising 60,887 individual character images that have been identified and segmented by hand, the collection represents the fruit of considerable human labor. It is suitable for use both in tasks specific to the Syriac language such as character recognition and manuscript dating, as well as for more general pattern recognition applications. Indeed, the registered version of the data offers a task that amounts to a larger, more complex version of the well-known MNIST digit recognition task.

In addition to the introduction of the data itself, this paper has looked at several applications that explore its properties. These include character recognition, character segmentation, and manuscript dating. Since the data are new there is no existing work to compare against, but it is hoped that the results presented herein will provide a baseline for measuring advances of the future.

REFERENCES

[1] S. Brock. 2012. A Tentative Checklist of Dated Syriac Manuscripts up to 1300. *Journal of Syriac Studies* 15, 1 (2012), 21–48.
[2] B. Efron and T. Hastie. 2016. *Computer Age Statistical Inference:Algorithms, Evidence and Data Science.* Cambridge University Press.
[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Generative Adversarial Nets. NIPS 2014: 2672-2680 Bengio. 2014. Generative Adversarial Nets. In *Neural Information Processing Systems.* 2672–2680.
[4] S. He, P. Samara, J. Burgers, and L. Schomaker. 2016. Historical manuscript dating based on temporal pattern codebook. *Computer Vision and Image Understanding* 152 (2016), 167–175.
[5] S. He, P. Samara, J. Burgers, and L. Schomaker. 2016. Image-based historical manuscript dating using contour and stroke fragments. *Pattern Recognition* 59 (2016), 159–171.
[6] S. He, P. Samara, J. Burgers, and L. Schomaker. 2016. A multiple-label guided clustering algorithm for historical document dating and localization. *IEEE Trans. on Image Processing* 25 (2016), 5252–5265.
[7] N. Howe and S. Xie. 2017. Chronological Profiling for Paleography. In *Int. Conf. on Document Analysis and Recognition.* to appear.
[8] N. Howe, A. Yang, and M. Penn. 2015. A Character Style Library for Syriac Manuscripts. In *Proceedings of the 2015 Workshop on Historical Document Imaging and Processing.* ACM.
[9] Nicholas R. Howe. 2013. Document Binarization with Automatic Parameter Tuning. *Int. J. on Document Analysis and Recognition* 13, 3 (September 2013), 247–258. DOI: 10.1007/s10032-012-0192-x.
[10] G. Huang, V. Jain, and E. Learned-Miller. 2007. Unsupervised joint alignment of complex images. In *IEEE 11th. Int. Conf on Computer Vision.* 1–8.
[11] D. P. Kingma and J. Ba. 2014. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations.*
[12] A. L. Maas, A. Y. Hannun, and A. Y. Ng. 2013. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *Proceedings of the International Conference on Machine Learning.*
[13] M. Mirza and S. Osindero. 2014. *Conditional Generative Adversarial Nets.* Technical Report. arXiv. arXiv:1411.1784.
[14] F. Wahlberg, L. Mårtensson, and A. Brun. 2015. Large scale style based dating of medieval manuscripts. In *Proceedings of the 2015 Workshop on Historical Document Imaging and Processing.*
[15] F. Wahlberg, L. Mårtensson, and A. Brun. 2016. Large Scale Continuous Dating of Medieval Scribes Using a Combined Image and Language Model. In *Document Analysis Systems.*