https://doi.org/10.2352/issn.2168-3204.2022.19.1.08

Isolated Handwritten Character Recognition of Ancient Hebrew Manuscripts

Tabita L. Tobing, Sule Y. Yayilgan; Department of Information Security and Communication Technology, NTNU - Norwegian University of Science and Technology; Gjøvik, Norway.
Sony George; Department of Computer Science, NTNU - Norwegian University of Science and Technology; Gjøvik, Norway.
Torleif Elgvin; NLA University College; Oslo, Norway

Abstract

Character recognition is widely considered an essential factor in preserving and digitizing historical handwritten documents. While it has shown a significant impact, the character recognition of historical handwritten documents is still a challenging task. This work aims to present a study on building a character recognition system for a handwritten ancient Hebrew text utilizing convolutional neural networks, dealing with material degradation, script complexity, and varied handwriting style. Our research underlined the importance of creating a ground-truth dataset for a robust and reliable character recognition system. Moreover, this study compares the performance of four convolutional neural network models applied to our dataset.

Introduction

Character recognition (CR) is part of the pattern recognition technique that enables a machine or computer to automatically recognize the characters on a printed or handwritten document. CR has played an essential role in studying historical handwritten documents for decades, especially in document preservation and digitization. However, CR for historical handwritten documents still poses various technical challenges compared to printed or handwritten modern documents, such as material degradation (discoloration, stain, and different kinds of noises) and script complexity (overlapping and broken characters) [1]. Again, isolated handwritten character recognition is domain-specific due to the wide variation in the handwriting style of scribes, the type and quality of material and ink, and the overlap in characters. A prime example of a degraded historical handwritten document is The Great Isaiah scroll, as provided in Figure 1. The Great Isaiah Scroll is one of the most intact Dead Sea Scrolls, a ”collection” of writings inscribed in ancient Hebrew scripts, the scroll dates from ca. 100 BCE. To automatically recognize individual or isolated characters in the scrolls is another challenge since studies show that two scribes wrote the Great Isaiah scroll or 1QIsa^a [2]. This means that we must deal with varied handwriting styles to recognize each character of the 22 ancient Hebrew characters.

This work addresses the challenges of isolated handwritten character recognition of the Great Isaiah Scroll and investigates the performance evaluation of several types of convolutional neural network architectures. This has not been done for the Great Isaiah Scroll to the best of our knowledge. Moreover, we have created an isolated ancient Hebrew characters dataset using image preprocessing approaches and manual segmentation and labeling.

Related Works

There has been much great literature showing the great success of using convolutional neural networks for handwritten character recognition [3]. For example, Rabby et al. [4] proposed a multilayer CNN for classifying Bangla handwritten characters on the available Bangla characters dataset and successfully tested the network on an entirely different Bangla dataset with 95.01% accuracy. Shalaka et al. [5] have developed a fine-tuned VGG-16 network and created a new dataset for Devanagari handwritten character recognition. They improved testing accuracy (from 85.64% to 93.17%) after using their proposed network compared to pre-trained VGG-16. Using 11-layers convolutional neural networks, Yue et al. [6] achieved 94.66% testing accuracy on the ancient Chinese handwritten characters dataset. There are other CNN-based handwritten character recognition systems with a testing accuracy of more than 90% for different languages: Japanese [7], Urdu Nastaliq [8], Arabic [9], and many more. There is one publicly available dataset created by Irina et al. [10] for the Hebrew language. This dataset consists of isolated cursive Hebrew handwritten characters with training and testing data distribution. They applied various CNN models such as simple CNN, AlexNet, and ResNet on the dataset and achieved testing accuracy of 72,57%, 78,21%, and 84,9%, respectively.

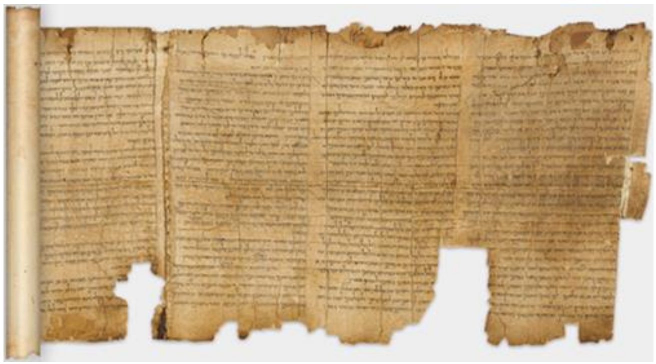

Figure 1. *Digital image of the Great Isaiah Scroll Column I-IV. Image source: The Israel Museum, Jerusalem. Accessed via http://dss.collections.imj.org.il/isaiah, used with permission.*

To the best of our knowledge, there has not been any available dataset for the isolated character of ancient Hebrew script. There is no study on isolated ancient Hebrew handwritten character recognition that in detail examines the performance of CNN models and applies it to separate testing data.

Ancient Hebrew Script

Through hundreds of years of ancient periods, the ancient Hebrew script took various forms due to variations in the writing style of writers who existed at different times and in different environments. Therefore, recognizing isolated characters from one specific ancient Hebrew manuscript is highly recommended.

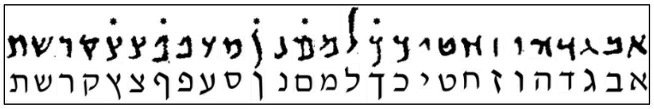
אבגדהוזחטיכךלמםנןסעפףצץקרשת

__Figure 2.__ The 27 characters of Hebrew alphabets. 1QIsa^a script: top row. The modern printed Hebrew script: bottom row. The • sign indicates the final letters.

To know how the specified ancient Hebrew script differs from the modern Hebrew is depicted in Figure2. It shows examples of 27 characters of the Hebrew alphabet, and five of them are final letters, the shape of the regular letter (to the right of the final letters) when they are written at the end of a word. Unlike the modern Hebrew script, the ancient script in 1QIsa^a only consists of consonants, without vowels. Additionally, some of the characters from the 1QIsa^a script appear to have a different shape compared to the modern Hebrew script.

Limitations

1QIsa^a scroll is an ancient Hebrew manuscript that is considered the most intact and longest scroll in the Dead Sea Scrolls collection, a perfect candidate for dataset construction for the machine learning-based CR system. However, this scroll also brings some challenges to implementing the CR system. The challenges are material degradation, inconsistency in using the final letter, and the presence of two hands in one scroll. In this study, we will not analyze the material degradation in 1QIsa^a but instead focus on applying image pre-processing to increase the legibility of the script. The second challenge is regarding the use of final letters; the authors of 1QIsa^a apply the use of final letters inconsistently also the forms of the final letters ן , ף , andץ have a shape similar to the standard form. Therefore, we will not include final letters for this study but only focus on 22 regular letters. For the last challenge, two scribes wrote 1QIsa^a scroll, i.e., the first scribe's handwritings in Column I-XXVII and the second scribe's handwritings in Column XXVIII-XLIV. With the variation of writing that is obvious between the two hands, this study will only use characters from the first scribe's handwriting.

Proposed System

The flowchart for handling the isolated handwritten character recognition, as shown in Figure 3, follows general major stages in character recognition [11].

Ground Truth Dataset

The digital images we used are color images of the Great Isaiah Scroll from the 1972 edition of John Trever's photographs (taken in 1948). It has CC BY-ND 4.0 license and is publicly available on the Internet Archive website [12]. The reason for creating a ground truth dataset is because there is no available isolated handwritten character of ancient Hebrew script dataset, and there is the need to introduce the CR system to accurate data, which can then be used for real-world scenarios. To build a reliable dataset, a philological expert of the Dead Sea Scrolls study is participating in inspecting the criteria for the characters to be extracted.

Image preprocessing

The main obstacle observed when extracting text from the ancient handwritten text was smears and uneven background. These two things result in poor legibility of the text. Therefore, image pre-processing is needed to increase the legibility of the text.

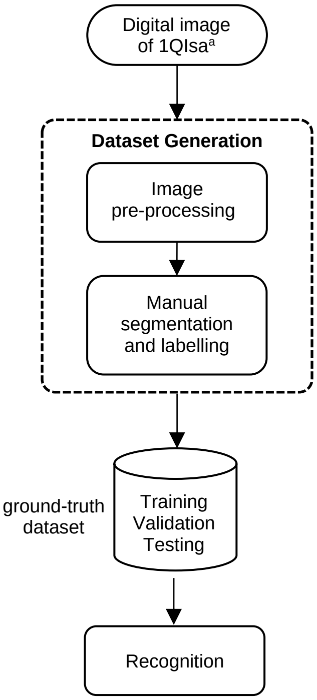

__Figure 3__. Flowchart of the isolated handwritten character recognition of ancient Hebrew manuscripts.

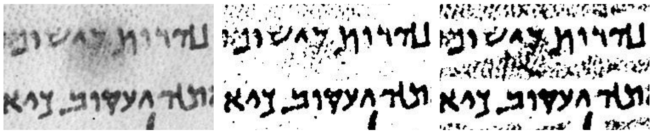

__Figure 4.__ After pre-processing. Grayscale: left column. Local thresholding with closing: middle column. Local Otsu with closing: right column.

To get a clear image of an isolated character. First, we need grayscale conversion of the color images before extracting isolated characters. The grayscale representation is necessary since it provides less information for each pixel, reducing computational requirements. Following this, image thresholding with morphological closing was applied to improve text legibility. In Figure 4, it is shown that local Otsu thresholding and morphological closing can separate characters from text background better than local thresholding with closing. Although this approach has not been able to remove the background color completely, this approach is good enough to remove the background color from around the text line. This result is enough to prepare the text for the subsequent manual segmentation and labeling stage.

Manual segmentation and labeling

Isolated character images were obtained with the help of python-based labeling software, LabelImg (Fig. 5). First, we need to create a rectangular box and put it on the area surrounding the letter we want to extract from the text. Afterward, the software will display a window where we need to input classes of the letters that we cut earlier. We have collected 3960 isolated characters of ancient Hebrew script from the 1QIsa^a scroll, with 180 characters for each of the 22 letters (see Fig. 6). Although it appears in Figure 6 that the letters are pretty clear, in fact, there are many images of single letters

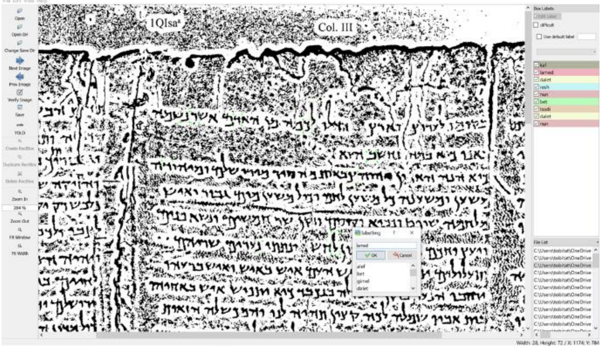

Figure 5. *Software LabelImg for isolated character extraction.*

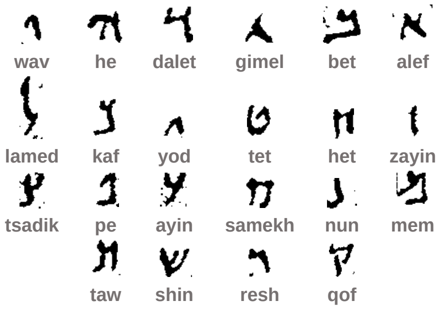

Figure 6. *The 22 isolated letters and their corresponding name.*

that are damaged (not intact), still have a background color and a small part from the letters next to them (compare Fig.6 with Fig.7).

CNN Models

CNN models have proven their remarkable performance in handwritten character recognition, not only limited to one language but many. This study used four types of CNN models that are widely used in character recognition system, i.e., LeNet-5, AlexNet, VGG16, and ResNet50.

LeNet-5

LeNet-5 is a CNN architecture intended for machine learning-based handwritten character recognition, introduced by LeCun et al. [13] in 1998. It comprises seven layers containing trainable parameters: a convolutional layer with six feature maps, a subsampling with six feature maps, a convolutional layer with 16 feature maps, a subsampling layer with 16 feature maps, a convolutional layer with 120 feature maps, one fully connected layer with 84 units, and the output layer corresponding to the number of classes. The input size of LeNet-5 is a 32x32 pixel image because it was initially intended for the MNIST database [14]. However, this architecture can also be used for larger images, keeping everything else constant. This is possible because the 5th layer is a convolutional layer with a feature map size of 1x1, not a fully connected layer.

AlexNet

AlexNet is an eight with-weight-layers CNN architecture introduced by Alex et al. in 2012 [15]. It consists of five convolutional layers, three subsampling layers, and three fully-connected layers. The input size of AlexNet should be 227x227x3, not 224x224x3 as said in the original paper, since the 227-size fits the math of the architecture. AlexNet has three consecutive layers of convolutional network, making it unique compared to other typical CNN architectures.

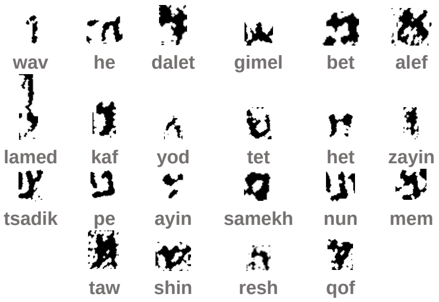

Figure 7. *Samples of letters in the dataset with damaged shapes, background noise, and some part of another letter.*

VGG16

Having the same filter and stride size for each convolutional layer and having padding and max pool layer with the same filter and stride size are the main characteristics of VGG16 architecture. VGG16, the 16 layers deep of a convolutional neural network, was proposed by Simonyan et al. in 2014 [16]. It consists of 13 convolutional layers, five subsampling layers, two fully connected layers, and one output layer. The input to this architecture is a 224x224x3 image. This study used pre-trained VGG16 on the ImageNet database [17].

ResNet50

ResNet50 is a convolutional neural network 50 layers deep. It was introduced by He et al. in 2016 [18]. It consists of 48 convolutional layers and two subsampling layers. The default input size of this network is 224x224x3. Similar to VGG16, ResNet50 has a pre-trained version trained on the ImageNet database. This study used the pre-trained version of ResNet50.

Training, Validation, Testing

Data used for character recognition is constructed as follows: training (60%), validation (20%), and testing (20%). Validation data is created to see the performance of CNN models training to avoid overfitting and underfitting. Then, CNN models were applied to the testing dataset, a dataset that has never been used in training. Before the isolated character enters the training stage, each of the isolated character images is sized according to the input requirements of each CNN model. The size of each image varies greatly, such as 19x24 for the letter with the smallest size, 27x97 for the tallest letter, and 37x46 for the letter with the broadest size. To overcome these size variations, zero padding is applied to each image until it reaches a dimension of 100x100. Then each image is resized to meet the default input size requirement of AlexNet, VGG16, and ResNet50.

Result and Discussion

Several CNN models were trained, validated, and tested on a new isolated ancient Hebrew handwritten character. In each model, batch normalization and dropout algorithms are embedded. Batch normalization was used to normalize input values from one to the next fully-connected layer. The dropout rate of 0,5 was applied to prevent overfitting. We also included an early stopping algorithm to avoid overfitting and underfitting in each model. The algorithm monitors the loss value of the validation set with patience = 10. If there are ten epochs where the training does not experience improvement in the validation loss value (improvement means that the value drops to zero), the training will stop. During training, another thing to consider is the variation of the learning rate optimizer for each model. The learning rates applied in each model are 0.01, 0.001, 0.0001, and 0.00001. The learning rate for optimal model performance can be seen in Table 1. We avoid using a learning rate that is too high or too low. A learning rate that is too high will make the weight updates too broad so that the validation graph will appear to be oscillating. Meanwhile, a learning rate that is too low makes the model get stuck at the same loss value.

Table 2 shows the length of time required per epoch and the total training time for each model. If sorted from the fastest total training time, the order is LeNet-5, AlexNet, ResNet50, then VGG16 with a total time in minutes, approximately: 6 minutes, 28 minutes, 58 minutes, and 1 hour 25 minutes. Total parameters respectively:3 million, 58 million, 26 million, and 134 million.

Table 3 presents the accuracy and loss of the training and validation set. It is shown that LeNet5 and AlexNet have slight differences in accuracy value between the training and validation set: 2,16% for LeNet-5 and 2,53% for AlexNet. On the other side, VGG16 and ResNet50 possess pretty wide gaps in the accuracy of the training and validation set: 9,58% and 10,69%, respectively. Therefore, VGG16 and ResNet50 can be perceived as having the possibility of overfitting.

Table 1. Hyperparameters of CNN Models

Model	Epoch	Optimizer
LeNet-5	27	Adam (learning_rate = 0,001)
AlexNet	20	Adam (learning rate= 0,0001)
VGG16	16	Adam (learning rate = 0,0001)
ResNet50	28	Adam (learning rate = 0,0001)

Table 2. Training Time of CNN Models

Model	Time per Epoch (s)	Total (s)
LeNet-5	± 7	± 189
AlexNet	± 84	± 1680
VGG16	± 322	± 5152
ResNet50	± 125	± 3500

Table 3. Training and Validation Performance

Model	[Accuracy	Loss]		
	Training		Validation	
LeNet-5	99,64%	0,028	97,48%	0,091
AlexNet	96,69%	0,105	94,16%	0,185
VGG16	99,49%	0,021	89,91%	0,391
ResNet50	99,96%	0,003	89,27%	0,354

Table 4. Testing Performance and TVT Standard Deviation

Model	[Accuracy	Loss]		
	Testing		TVT Standard Deviation	
LeNet-5	96,34%	0,125	1,369	0,04
AlexNet	94,44%	0,158	1,132	0,033
VGG16	91,79%	0,352	4,144	0,166
ResNet50	89,39%	0,334	5,011	0,161

Table 5. Most Confused Letters

LeNet-5	AlexNet	
Dalet (30 TPs out of 36 samples) 	*Pe* (28 TPs out of 36 samples) 	
Confused With	Confused With	
5x 	5x 	1x
1x 	1x 	1x

For CNN models' performance in predicting images in the testing set is presented in Table 4. The most reliable CNN model based on the training-validation-testing (TVT) standard deviation goes to AlexNet, followed by LeNet-5. VGG16 and ResNet50 hold standard deviation values about four-time higher than AlexNet and LeNet-5, indicating poor performance in recognizing the letters.

Table 5 presents the most confused letters derived from LeNet-5 and AlexNet training. LeNet-5 predicted *Dalet* with 30 true positives (TPs), and the rest were predicted as letter *Resh* (five times) and *He* (once). In AlexNet, the letter Pe was predicted with 28 TPs, and the rest were predicted as letter *Nun* (five times), letter *Tet* (once), letter *Bet* (once), and letter *Taw* (once).

Conclusion and Future Works

Recognizing isolated letters in the ancient handwritten text is a complex task. In that sense, we need various considerations in deciding the computational approach that fits the existing data. In this work, we have investigated the challenges to the CR system on the Great Isaiah Scroll images. We have created a new dataset, a collection of images of isolated letters found in this scroll. We also successfully compared the performance of four CNN models applied to our dataset. The findings suggest that AlexNet and LeNet-5 are the most suitable CNN models for isolated ancient Hebrew handwritten character recognition with narrow TVT standard deviations of accuracy and loss rate and more than 94% testing accuracy.

To further our research, we intend to expand our dataset with isolated letters from the handwriting of the second scribe of the 1QIsa^{a} scroll. Also, we are building an automated handwritten character recognition, especially for the 1QIsa^{a} scroll.

Acknowledgment

This work is supported by The Lying Pen of Scribes−Manuscript Forgeries, Digital Imaging, and Critical Provenance Research funded by the Research Council of Norway (projectnr. 275293).

References

[1] N. Babu and S. A., "Character Recognition in Historical Handwritten Documents – A Survey," 2019 International Conference on Communication and Signal Processing (ICCSP), pp. 0299-0304, 2019.

[2] E. Tov, Scribal Practices and Approaches Reflected in the Texts Found in the Judean Desert. Leiden: Brill, 2004, 21.

[3] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," IEEE Access, vol. 8, pp. 142642-142668, 2020.

[4] AKMSA. Rabby, S. Haque, S. Abujar, SA. Hossain, “EkushNet: Using Convolutional Neural Network for Bangla Handwritten Recognition,” Procedia Computer Science, vol. 143, pp. 603-610, 2018.

[5] SP. Deore and A. Pravin, “Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset,” Sādhanā 45, 243, 2020.

[6] Y. Xu, F. Yin, D. -H. Wang, X. -Y. Zhang, Z. Zhang and C. -L. Liu, "CASIA-AHCDB: A Large-Scale Chinese Ancient Handwritten Characters Database," 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 793-798, 2019.

[7] C. Tsai, "Recognizing handwritten Japanese characters using deep convolutional neural networks." University of Stanford, California, 2016.

[8] S. Naz, AI. Umar, R. Ahmad, I.Siddiqi, SB. Ahmed, MI. Razzak, and F. Shafait, “Urdu Nastaliq recognition using convolutional–recursive deep learning,” Neurocomputing, vol. 243, pp. 80-87, 2017.

[9] KS. Younis, "Arabic handwritten character recognition based on deep convolutional neural networks,” Jordanian Journal of Computers and Information Technology (JJCIT) 3.3, pp. 186-200, 2017

[10] I. Rabaev, BK. Barakat, and A. Churkin, "The HHD dataset," 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE, 2020.

[11] N. Arica and F. T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 31, no. 2, pp. 216-233, 2001.

[12] JC. Trever, FM. Cross, DN. Freedman, and JA. Sanders, “Scrolls from Qumrân cave I; the great Isaiah scroll, the Order of the community, the Pesher to Habakkuk”, Albright Institute of Archaeological Research and the Shrine of Book, Jerusalem, 1974, accessed 22 November 2021, <https://archive.org/details/qumran/page/n15/mode/2up>.

[13] Y. LeCun, L. Bottou, Y. Bengjo, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE 86.11, pp. 2278-2324, 1998.

[14] L. Deng, The mnist database of handwritten digit images for machine learning research,” IEEE Signal Processing Magazine, 29(6), pp. 141–142, 2012.

[15] A. Krizhevsky, I. Sutskever, and GE. Hinton. "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems 25, 2012.

[16] K. Simonyan and A Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[17] J. Deng, W. Dong, R. Socher, LJ. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” IEEE conference on computer vision and pattern recognition, pp. 248–255, 2009.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

Author Biography

Tabita A. M. L. Tobing received her BSc in Physics from Universitas Indonesia (2016) and MSc degree in Electrical Engineering from Institut Teknologi Bandung (2021). She is currently a PhD student at NTNU under The Lying Pen of Scribes project; her research field is the computational analysis of ancient Hebrew manuscripts using machine learning.

Sule Y. Yayilgan is currently a Professor at the Department of Information Security and Communication Technology (IIK), NTNU. She belongs to the Center for Cyber Information Security (ccis.no), and she is leading the research group MR PET: Multidisciplinary Research group on Privacy and data protEcTion. Her research interests include Artificial Intelligence and Cybersecurity in various application fields.

Sony George is currently Associate Professor at The Colourlab, NTNU, since 2017. Before joining NTNU, he worked as a researcher at Gjøvik University College, Norway. Sony obtained a PhD in Photonics from the Cochin University of Science and Technology, India, in 2012. His research interests are in color imaging, spectral image processing, computer vision, etc.

Torleif Elgvin is currently Professor Emeritus of theology, religion, and philosophy at NLA University College. He holds a theological degree from MF Norwegian School of Theology, Religion, and Society. He received his Doctorate on the Dead Sea Scrolls at the Hebrew University of Jerusalem in 1998. He was head of the Nordic network for Dead Sea Researchers from 2003 to 2007 and, since 1992, has participated in the publishing team for the Dead Sea Scrolls. His research interests are Dead Sea Scrolls and biblical studies.