Ingénierie des Systèmes d'Information
Vol. 29, No. 6, December, 2024, pp. 2243-2250
Journal homepage: http://iieta.org/journals/isi

A Comparative Study of Four Handwritten Text Recognition Models in Arabic Script

Feras Aljishi[1], Raed Mughaus[1,2], Hamzah Luqman[1,2], Mohammad Tanvir Parvez[3*]

[1] Department of Information and Computer Science, King Fahd University of Petroleum & Minerals (KFUPM), Dhahran 31261, Saudi Arabia
[2] SDAIA-KFUPM Joint Research Center for Artificial Intelligence, Dhahran 31261, Saudi Arabia
[3] Department of Computer Engineering, College of Computer, Qassim University, Buraydah 51477, Saudi Arabia

Corresponding Author Email: m.parvez@qu.edu.sa

https://doi.org/10.18280/isi.290614

Received: 4 January 2024
Revised: 4 September 2024
Accepted: 25 September 2024
Available online: 25 December 2024

Keywords:
Arabic handwriting recognition, handwritten text recognition, neural networks, pattern recognition, KHATT

ABSTRACT

Handwritten text recognition (HTR) is the technique of recognizing and interpreting handwritten text into machine-readable output. HTR is a challenging problem given the variance in handwriting styles across people and the poor quality of the handwritten text. However, considerable work has been accomplished to recognize Latin scripts. In contrast, the accuracy of Arabic HTR systems is far behind the HTR of Latin script. In this paper, a comparative experimental assessment of four recent deep learning models (namely, FCN, GFN, VAN, and DAN) that have been proposed for HTR of Latin scripts. These models are evaluated on the KHATT dataset, a challenging Arabic handwritten text dataset. The lowest CER and WER are obtained using the DAN model. In addition, a deep analysis of the challenges related to the Arabic HTR is discussed.

1. INTRODUCTION

Arabic, a language spoken by over 400 million people in Arabic countries, is one of the six official languages of the United Nations. Its unique script is also used in several other languages, including Kurdish, Persian, Pashto, and Urdu. Additionally, Arabic is the language of the Qur'an, a sacred text read by countless Muslims worldwide.

With a cursive style and a right-to-left writing direction, Arabic script has letters that can vary in appearance based on their position within words. The basic alphabet consists of 28 letters, but some researchers consider four additional ligatures, increasing the total to 36 or 40 [1].

Optical character recognition (OCR) is the automatic reading of the information and converting it into an electronic form [2]. The information is scanned from paper documents "offline" or input using touch screens and other devices "online". OCR has a wide variety of applications. It helps in converting handwritten text into computer-editable text to be used by search engines. It also helps in extracting the handwritten text from historical documents.

Offline handwritten text recognition (OHTR) has recently attracted more interest due to the advances in image-capturing devices (e.g., smartphones). Offline text is represented as a text image. Unlike offline handwriting, online text is composed of a series of strokes. Each stroke is defined by a set of points that indicate its spatial location and the timing or pressure applied during writing. This information is not available with offline text images which makes OHTR more challenging than online handwritten text recognition.

Arabic OHTR has seen the development of various techniques, which can be categorized as classic and machine learning methods. Classic techniques depend on hand-crafted features extracted from the handwritten text and fed usually into machine learning classifier [3-6]. Recently proposed approaches depend mainly on machine learning for automatically learning and classifying the features from the handwritten text [7]. However, the reported results of these systems are far to be commercialized compared with OHTR of non-Arabic languages such as English and Chinese languages.

Figure 1 shows the framework of the OHTR system. As shown in the figure, the digitalized image is fed into the OHTR to recognize the written characters in order to recognize the whole text. In this work, we evaluate four state-of-the-art OHTR models, that have been for proposed for Latin script HTR, for Arabic script HTR. These models are fully convolutional network (FCN) [8], gated fully convolutional network (GFCN) [9], vertical attention network (VAN) [8], and document attention network (DAN) [10].

These models have been evaluated on the KHATT dataset which consists of a challenging Arabic handwritten text written by several writers. In addition, we evaluated the effectiveness of different preprocessing techniques for reducing the character and word error rates.

This paper is divided into five sections. Section 2 reviews the literature on OHTR. Section 3 details the evaluated models. Section 4 presents the experimental results. And finally, Section 5 concludes the paper.

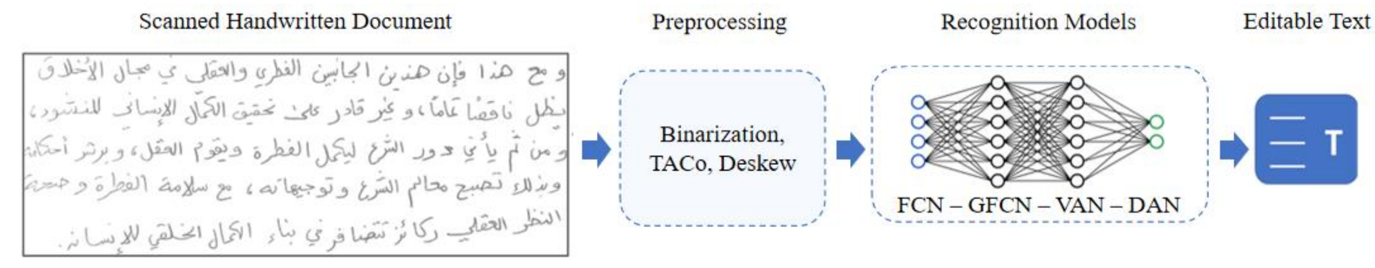


Figure 1. The framework of an OHTR system

2. RELATED WORKS

In this section, there are a lot of research works in HTR that focus on deep learning methods [11-13]. We discuss methods that target Arabic and other languages.

Singh and Karayev [14] presented a handwritten text recognition neural network model based on Image-to-Sequence architecture. The encoder was a CNN that extracts a 2D feature map from an image. The decoder was a transformer stack. The datasets used were IAM, WikiText, Free Form Answers, Answers2 & Names. This model performed better than all the commercial cloud APIs. Combined with paragraph segmentation, the model achieved a 5.5 CER score on IAM. For single-line recognition, it achieved a 4.4 CER score. Wang et al. [15] used DAN (Decoupled Attention Network) for an end-to-end text recognition model designed to handle irregular text. This model consists of a feature encoder, a convolutional alignment module, and a decoupled text decoder. Tested on IAM and RIMES datasets, DAN demonstrated strong performance in both regular and irregular text recognition, achieving a 6.4 CER on IAM and a 2.7 CER on RIMES.

Li et al. [16] introduced TrOCR. TrOCR is an end-to-end text recognition model based on decoder- encoder architecture. TrOCR utilizes pre-trained transformers in both the encoder and decoder components. Multiple handwritten and printed text datasets were used, including IAM. TrOCR outperformed the state- of-the-art handwritten text recognition models, with a 2.89 CER score. Yousef et al. [17] proposed a handwritten text recognition model based on a fully convolutional neural network without recurrent connections. Multiple datasets were used, including KHATT and IAM. It achieved an 8.7 CER score on KHATT and a 3.5 C@6 score on IAM. This architecture won the ICFHR2018 Competition on Automated Text Recognition.

Wigington et al. [18] introduced Start, Follow, Read (SFR) model. SFR is a deep learning model that performs text detection, segmentation, and recognition. SFR consists of three neural networks: one network to find the positions of text lines, another that processes and dewarps text lines into images, and a CNN-LSTM recognition network that takes these dewarped images as input. SFAR has been tested with IAM & RIMES datasets. SFR achieved a 6.4 CER score on IAM and a 2.1 CER score on RIMES. Coquenet et al. [8] proposed an end-to-end handwritten text recognition model based on a vertical attention network. The model has been evaluated on three datasets: RIMES, IAM, and READ 2016. It achieved state-of-the-art results, a 1.91 CER score on RIMES, a 4.45 CER score on IAM, and a 3.59 CER score on READ 2016.

Coquenet et al. [19] introduced Simple Predict & Align Network (SPAN). SPAN is an end-to-end handwritten text recognition model based on a fully convolutional network without recurrent connections. SPAN is simple and does not require segmentation. SPAN performed very well on RIMES, IAM & READ 2016 datasets. It achieved a 4.17 CER score on RIMES, a 5.45 CER score on IAM, and a 6.2 CER score on READ 2016. Coquenet et al. [9] proposed end-to-end text recognition model based on a recurrence-free, fully gated convolutional neural network architecture. When tested with RIMES, IAM & READ 2016 datasets, the model showed competitive results. It achieved a 4.35 CER score on RIMES, and a 7.99 CER score on IAM.

Chowdhury and Vig [20] proposed an end-to-end text recognition model that combines a CNN and an encoder-decoder network. The encoder-decoder network uses LSTM cells with 256 hidden units. The used datasets are IAM & RIMES. The model achieved an 8.1 CER score on IAM and a 9.6 CER score on RIMES. Chaudhary and Bali [21] presented Efficient and Scalable Text Recognizer (EASTER). EASTER is a text recognition model based on a recurrence-free neural network with 1-D convolutional layers. While most recent research proposed complex solutions that utilize recurrent networks or gated layers, EASTER model is simple and more efficient. The used dataset is IAM. It achieved a 7.9 CER score.

Based on memory-augmented neural networks (MANNs), Nguyen et al. [22] presented Convolutional Multi-way Associative Memory (CMAM). CMAM is a text recognition model architecture that leverages recent memory accessing techniques in MANNs. Compared with Convolutional Recurrent Neural Net- works, CMAM showed superior performance on long text datasets. The datasets used are IAM, SCUT-EPT, and Private. CMAM achieved an 11.12 CER score on IAM and a 10.71 CER score on SCUT-EPT. Chung and Delteil [23] introduced a computationally efficient handwritten text recognition framework. The framework consists of multiple models: 1) a handwritten text detection neural network, 2) a multi-scale convolutional neural network that extracts features from detected regions, and 3) a bidirectional LSTM network. While using less memory and time, this model achieves outstanding results. The dataset used is IAM. It achieved an 8.5 CER score on IAM.

Kang et al. [24] proposed a recurrence-free approach that utilizes transformer models. Multi-head self- attention layers are being used for both the visual and textual stages. The dataset used is IAM. The proposed model achieved a 4.67 CER score. Such et al. [25] proposed a text recognition model based on a fully convolutional neural network. An attention-based technique has been introduced to handle the variety of handwriting with different stroke widths, slants, and noise. The datasets used were IAM & RIMES. The proposed model achieved a 4.43 CER score on IAM and a 2.22 CER score on RIMES.

Kass and Vats [26] proposed an attention-based sequence-to-sequence model for HTR. The encoder consists of two stages: ResNet & bidirectional LSTM, and the decoder uses a constant-based attention mechanism. The datasets used are Imageur5K and IAM. This model achieved a 6.5 CER score on the IAM dataset. Wick et al. [27] proposed using CTC-Prefix-Score during the decoding stage of the sequence-to-sequence HTR models. The model architecture consists of a CNN visual backbone, a bidirectional LSTMs encoder, and a Transformer decoder. The model has been evaluated on IAM, Rimes, and StAZH datasets. This model achieved a 3.13 CER score on the IAM dataset.

Bhunia et al. [28] aimed to develop a single model for HTR and scene text recognition (STR). A knowledge distillation (KD) based framework has been introduced. The model has been trained on multiple datasets, including IAM and RIMES. Quantitative performance against other models showed that the model achieved 86.4% on the IAM dataset. Inspired by the presented differentiable attention models for speech recognition, image captioning, and translation, Bluche et al. [29] proposed an attention-based model for end-to-end handwriting recognition. While previous HTR models had required line segmentation, this model was the first to perform end-to-end multi-line handwriting recognition. The dataset used is IAM. The model achieved a 9.4 CER score on IAM.

Bluche [30] modified the MDLSTM-RNNs architecture for end-to-end HTR, replacing the collapse layer with a recurrent version for line-by-line recognition. The model uses attention weights on the image representation for implicit line segmentation. Evaluated on RIMES and IAM, it achieved a 5.5 CER on IAM and a 2.9 CER on RIMES. Graves and Schmidhuber [31] introduced an HTR model combining multidimensional recurrent neural networks and connectionist temporal classification. Tested on ICDAR 2007 Arabic handwriting, this model reached a 91.43% accuracy score.

3. METHODOLOGY

The framework of the Arabic OHTR system is described in this section. The general framework for the OHTR system is shown in Figure 1. This framework is almost similar for all pattern recognition systems including OHTR. However, some OHTR systems may have more details at every phase. The task of building an HTR system starts with preprocessing the data. In this phase, the dataset has been pre-processed for better training and recognition. This stage is followed by models training. Four models have been evaluated in this work for Arabic OHTR. These models have been trained and evaluated on the KHATT dataset.

3.1 Pre-processing

Arabic handwritten text is more challenging than typed text for OCR systems. This can be attributed to the nature of the handwritten text and to the problems that are inherited from the image-capturing devices. Therefore, there is a need to preprocess the text images before feeding them to the recognition models.

Several preprocessing techniques have been used in this work to prepare the text images for model training and evaluation (Figure 2).

We started the preprocessing stage by transforming the text images into binary images, where each pixel in the image can be either 0 or 1 only. Working with binary images is easier in the sense that many features used in various handwritten text recognition models depend on binary images [10]. Moreover, binarization process can remove some noises from the text image itself. The most common technique for image binarization is to use a threshold against a pixel value to decide whether that pixel value should be converted to 0 or 1. Binarization of a grey-scale image is a challenging problem [32], as different parts of a grey-scale image may require different thresholds for binarization. However, the grey-scale text images require only a single threshold for binarization since the image usually contains two parts, text and background. For this purpose, we used Otsu's method [32] for binarization. This method depends on finding the binarization threshold that maximizes the inter-class variance.

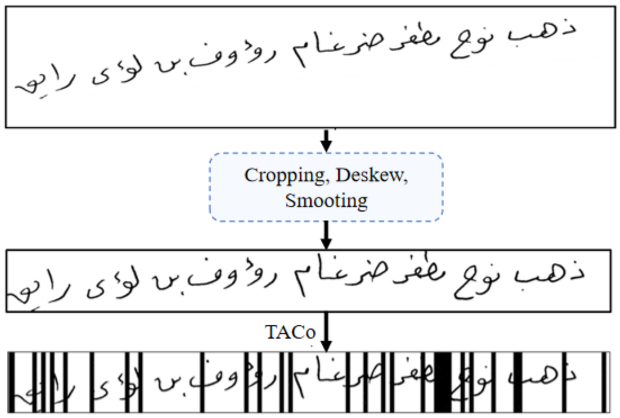


Figure 2. Preprocessing techniques used to prepare the data samples of KHATT dataset

KHATT dataset is available in two forms, paragraphs and lines. However, some of the dataset's lines were not segmented correctly. Many line samples of KHATT contain either large white blank spaces or pixels from other lines. To remove any extra pixels around the text that are not part of the text, we cropped these lines.

The resulting text lines from the cropping stage may be rotated and not horizontally aligned. This can result in some problems with recognition systems. To horizontally center the text lines, we deskewed the cropped text images. The main benefit of deskewing and straightening is that these operations can correct the baseline of the text image and make the baseline horizontally aligned, as shown in Figure 2. Several methods have been proposed in the literature for correcting the skew present in text image [2]. In this work, we rotated the image using horizontal projection. In this technique, we selected the highest peak of the projected line to be the new baseline of the text image. We also corrected the skew present in individual words by estimating the vertical strokes and their angles with the vertical axis.

The skewing preprocessing step is followed by text image smoothing. In this step, we used a 5 by 5 Gaussian filter. The purpose of this step is to remove the noise that may not be removed by the thresholding technique discussed before.

After preprocessing the dataset, we fed the training set into the data augmentation module. The goal here is to augment the dataset in order to enlarge its size and add more variability. This step is essential for many deep learning models that require a large quantity of samples for training. In this work, we employed Tiling and Corruption Augmentation (TACO) [33] for data augmentation.

TACO algorithm consists of two steps, tiling and corrupting. It starts by randomly segmenting the text image into many equal-sized tiles. Then, some of these tiles are replaced with the corrupted segments as part of the corruption step. As the final step, the tiles are stitched back together in the same sequence to form the augmented image [33]. Figure 2 illustrates the operation of TACo for a line of text.

3.2 OHTR models

In this work, we have evaluated four recent models for Arabic HTR. These models are the SOTA models for Latin script HTR. Each of these models accepts the pre-processed text image as an input and outputs the predicted text.

Gated Fully Convolutional Network: Gated Fully Convolutional Network (GFCN) was proposed by Coquenet et al. [9] as an alternative to CNN for spatial feature extraction and LSTM for sequence modeling. GFCN has several advantages over CNN-LSTM architecture, such as reducing the number of parameters using only convolutional components and parallel processing of the sequences to accelerate training. GFCN implements convolutional and pooling layers using a gating mechanism to imitate the behavior of LSTM. GFCN differs from Gated CNN by removing the dense layer, which enables GFCN to use input images of variable sizes [9]. Figure 3 shows the architecture of GFCN. As shown in the figure, GFCN consists of a sequence of convolution blocks followed by five gate blocks. Each gate block contains two Depthwise Separable Convolutional Blocks (DSCBs) to reduce the number of parameters.

Fully Convolutional Network (FCN): FCN used in this work is adapted from Coquenet et al. [8]. The architecture of this model is similar to GFCN without a gating mechanism. FCN consists of two components, encoder and decoder. The encoder module accepts text images with multiple text lines which makes this model appropriate for recognizing text images at the paragraph level. The encoder module consists of a sequence of Conventional Blocks (CB) and DSCB for feature extraction and reduction. The generated feature maps by the encoder are fed into the decoder. An implicit segmentation is applied in this model through the attention module that produces vertical attention weights to make the model focus on the features of the current line. Finally, the decoder module recognizes the whole paragraph by recognizing the characters of each line using its features. FCN offers several advantages, including end-to-end learning that optimizes the entire process from input to output, flexibility in handling varying input sizes, and preservation of spatial hierarchies essential for tasks such as segmentation [34]. However, FCNs also have some limitations. One of the primary drawbacks is their high computational cost and memory requirements, especially when dealing with high-resolution images or deep network architectures. This can limit their applicability in resource-constrained environments [35]. Furthermore, the fixed receptive field of convolutional layers can sometimes lead to inadequate context understanding for segmenting objects of varying scales [36].

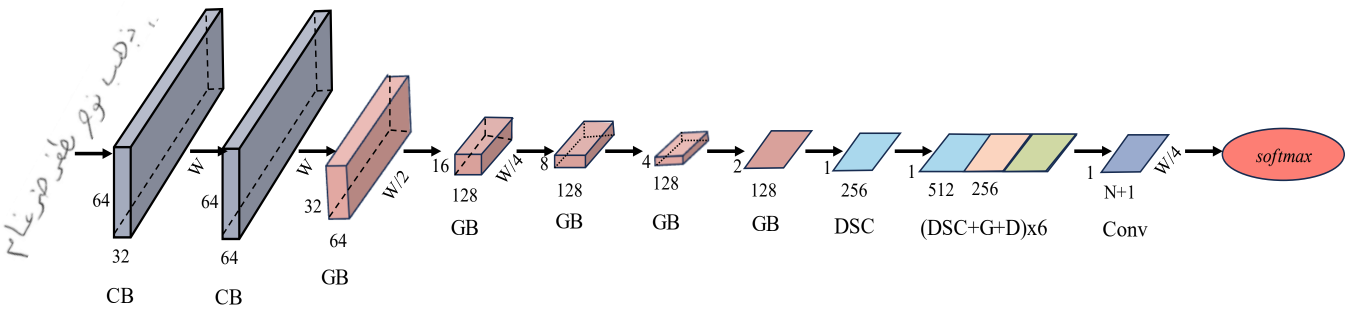


Figure 3. Illustration of GFCN architecture [9]

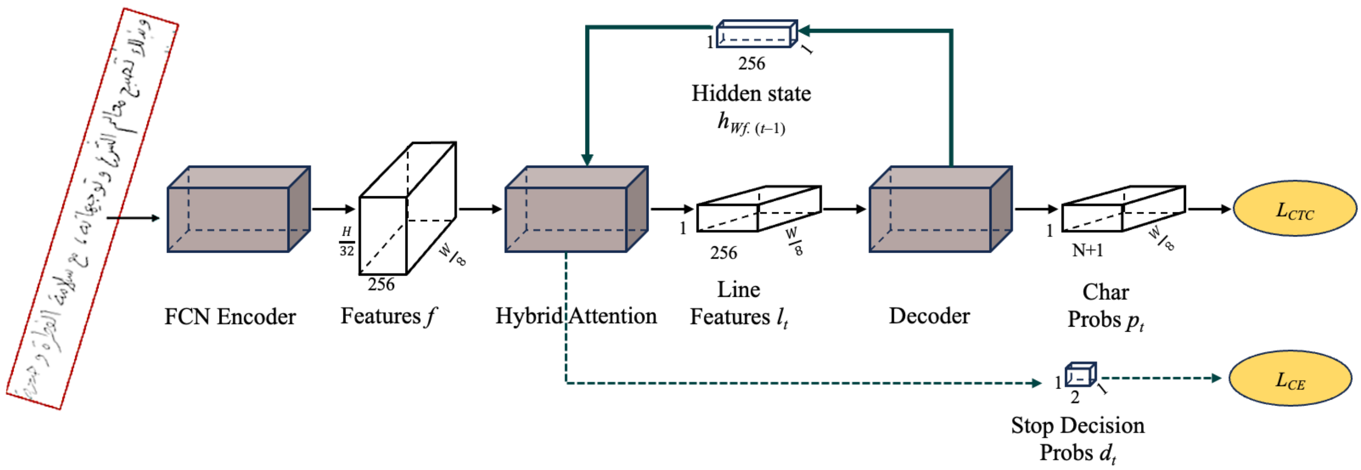


Figure 4. The general architecture of VAN model [8]

The FCN encoder takes as input an image of size (H, W, C) and outputs feature maps of size (H/32, W/8, 256). The encoder consists of 6 CBs and four DSCBs. The first three CBs reduce the width of the feature maps to W/8; each block divides the width by 2. At the same time, the first five CBs reduce the height of the feature maps. The number of feature maps produced by the first CB is 16, and each CB doubles the input feature maps count until reaching 256. Each CB consists of two convolutional layers, an instance normalization layer, and a convolutional layer. DSCB is similar to CB, except that it uses DSC. DSC is a more efficient type of convolution that consists of two main steps. First, a filter is applied to each input feature map to produce C feature maps, where C is the number of input channels. Then pointwise filters (filters of size 1x1xC) are applied to the output of the first step. The number of pointwise filters depends on the desired number of feature maps. The line decoder of FCN takes as input the features extracted by the FCN encoder. The decoder consists of adaptive max-pooling that reduces the height of the feature maps to 1 and a convolution layer that produces V feature maps, where V is the number of tokens. So, the output shape of the decoder is (1, W/8, V). The model weights are updated using the CTC loss.

Vertical Attention Network: Vertical Attention Network (VAN) uses an FCN encoder to deal with variable dimensions of images that contain multiple lines [8]. Then an attention module produces vertical attention weights that focus on the next feature of the following line. Then, the decoder recognizes the sequence of characters from the line features. One of the key advantages of VAN is its ability to focus on important features by leveraging attention mechanisms, which improves the accuracy of image classification and object detection tasks.

It effectively captures spatial hierarchies and dependencies, leading to a more robust feature representation. Moreover, VAN can handle large-scale data efficiently, making it suitable for real-world applications [8]. However, VANs often require substantial computational resources and training time, which can be a barrier for deployment in resource-constrained environments. Figure 4 illustrates the general architecture of VAN.

In the decoding phase of generating the recognized text, the model will keep recognizing a letter several times until the recognizer moves to the next letter. Depending on the repetition of each letter, VAN will decode each individual letter to generate the output word. The decoder will recognize if the letter is repeated in the input by generating an underscore symbol before the repeated letter, as shown in the example in Figure 5.

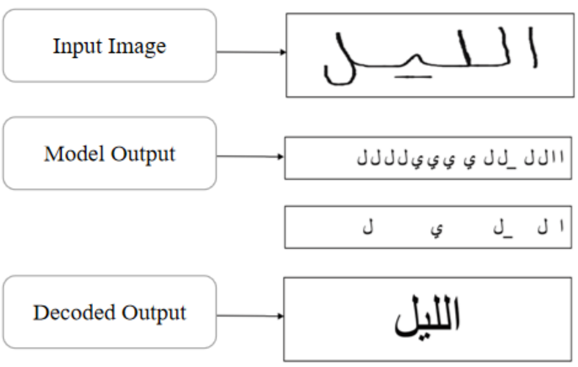
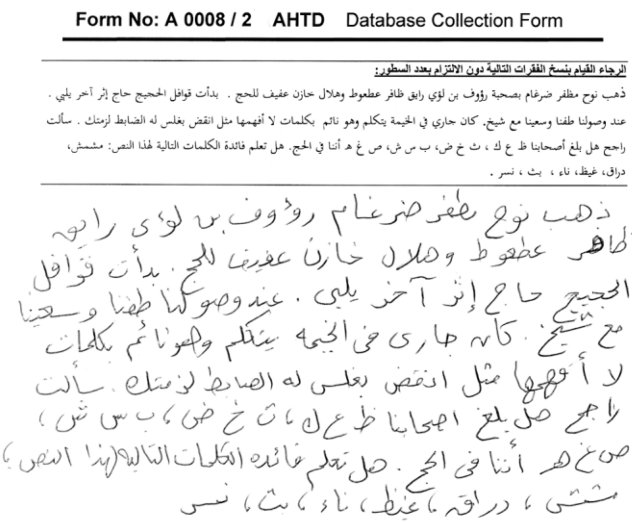


Figure 5. Process of word decoding in VAN

Document Attention Network (DAN): Document Attention Network (DAN) was introduced recently by Coquenet et al. [10] for Latin HTR. DAN flattens the output feature of an FCN encoder into a one- dimensional sequence of features. The decoder employs a recurrent process at the character level to predict tokens based on the calculated features. It generates token probabilities, and the token with the highest probability is selected as the final prediction. As a loss function, DAN uses a cross-entropy loss function.

4. EXPERIMENTAL RESULTS

In this work, we use KHATT (http://khatt.ideas2serve.net) dataset to evaluate the OHTR models. KFUPM Handwritten Arabic TexT dataset (KHATT) [37] is a highly comprehensive Arabic offline handwritten text dataset [37]. KHATT consists of 2,000 fixed-text paragraphs and 2,000 unique paragraphs on various topics written by 1,000 different writers. To ensure statistical representation, the text for the unique paragraphs in the dataset was randomly chosen from 46 different sources. In contrast, fixed paragraphs contain the same text written by all writers. To ensure some variations in the written text, the dataset contains also freely written text selected by the writers. In this work, we used the segmented lines of the unique paragraphs of the dataset for training and evaluating the proposed models. This part of the dataset was pre-divided into 70% training, 15% validation, and 15% testing sets. A sample of the KHATT dataset is shown in Figure 6.



Form No: A 0008 / 2 AHTD Database Collection Form

الرجاء القيام بنسخ الفقرات التالية دون الالتزام بعدد السطور:

ذهب نوح مظفر ضرغام بصحبة رؤوف بن لؤي رابق ظافر عطعوط وهلال خازن عفيف للحج . بدأت قوافل الحجيج حاج إثر آخر يلبي . عند وصولنا طفنا وسعينا مع شيخ. كان جاري في الخيمة يتكلم وهو نائم بكلمات لا أفهمها مثل انقض بغلس له الضابط لزمتك . سألت راجح هل بلغ أصحابنا ظ ع ك ، ث خ ض، ب س ش، ص غ ه أننا في الحج. هل تعلم فائدة الكلمات التالية لهذا النص: مشمش، دراق، غيظ، ناء ، بث ، نسر .

Figure 6. Sample data-form from KHATT dataset

We implemented the evaluated models using Pytorch. The models were trained with the CTC loss function and the Adam optimizer, set with a learning rate of 10−4, over 200 epochs. The VAN model, however, was trained for 5000 epochs. The character error rate (CER) and word error rate (WER) were used to evaluate the performance of the text recognition models. These metrics measure the amount of handwritten text that was incorrectly recognized. CER reports the percentage of characters that have been recognized incorrectly and WER is the percentage of incorrectly recognized words against the ground truth.

We began our experiments by assessing the effect of pre-processing techniques (described in Section 3-A) on models' performance. We selected the FAN model and applied each pre-processing technique independently before combining all of them and the obtained results are shown in Table 1. As shown in the table, the lowest error rates were obtained when combining all the processing techniques. However, the TACo technique was the most effective preprocessing technique at the character level since this technique motivates the model to learn the occluded characters making it efficient for the recognition of non-occluded characters.

Table 1. The performance of the fan model with different pre-processing techniques

Pre-processing	CER	WER
No pre-processing	0.130	0.514
Denoising	0.124	0.484
Deskew	0.126	0.492
TACO	0.116	0.486
All	0.113	0.475

Table 2. Recognition results of the evaluated models using KHATT dataset

Model	CER	WER
FCN	0.113	0.476
GFCN	0.166	0.602
VAN	0.107	0.439
DAN	0.089	0.376

Table 2 lists the CER and WER of the evaluated models. As can be seen in the table, the lowest CER and WER were obtained with the DAN model. The significant reduction was in the WER with around 6% compared with the VAN model. The GFCN model reported the highest error rates.

Although highly promising results were obtained using the DAN model, the results are still low compared to the state-of-the-art results reported for Latin script. This can be attributed to several reasons related to the Arabic script and the recognition models. Arabic script is challenging for recognition compared to other scripts, such as Latin. The difficulty source is mainly the cursive nature of Arabic script where most of the letters are connected to the previous and next letters. In addition, the shape of the character differs based on the letter location in the word. Moreover, some letters in the KHATT dataset were written in a vertical manner where more than one letter is written over each other as shown in Figure 7. This style of writing is difficult for segmentation and recognition.

Figure 7. Sample word from KHATT dataset with vertical letters

5. CONCLUSIONS AND FUTURE WORK

We present comparative experimental evaluations of four recently proposed deep learning models (FCN, GFN, VAN, and DAN) for HTR of Latin scripts. The assessment is conducted on the KHATT dataset, a challenging collection of Arabic handwritten text. The DAN model proves to be the most effective, yielding the lowest Character Error Rate (CER) and Word Error Rate (WER). Additionally, a comprehensive analysis of the challenges associated with Arabic HTR is presented in this paper.

The current work only focuses on offline Arabic handwriting recognition. The system was tested using KHATT database. The current work can be further validated with other databases on Arabic handwritten text.

The proposed system can be further enhanced by combining the different classifiers into a multi-classifiers system. Also, a system that can work for both online and offline text would be plus point. Moreover, the system should be tested against other languages (like Urdu, Parsian) that use scripts similar to Arabic script.

Another possible future work is to test the proposed system for digitization of scanned documents from some specific domain, like digitalization of scanned forms from banking sector [38].

ACKNOWLEDGEMENTS

Researchers would like to thank the Deanship of Scientific Research, Qassim University for funding publication of this research.

REFERENCES

[1] Habash, N.Y. (2010). Introduction to Arabic Natural Language Processing. Morgan & Claypool Publishers.

[2] Parvez, M.T., Mahmoud, S.A. (2013). Offline Arabic handwritten text recognition: A survey. ACM Computing Surveys, 45(2): 23. https://doi.org/10.1145/2431211.2431222

[3] Mezghani, N., Mitiche, A., Cheriet, M. (2002). On-line recognition of handwritten Arabic characters using a Kohonen neural network. In Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, Niagra-on-the-Lake, ON, Canada, pp. 490-495. https://doi.org/10.1109/IWFHR.2002.1030958

[4] Assaleh, K., Shanableh, T., Hajjaj, H. (2009). Recognition of handwritten Arabic alphabet via hand motion tracking. Journal of the Franklin Institute, 346(2): 175-189. https://doi.org/10.1016/j.jfranklin.2008.08.005

[5] Mezghani, N., Mitiche, A., Cheriet, M. (2008). Bayes classification of online Arabic characters by gibbs modeling of class conditional densities. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(7): 1121-1131. https://doi.org/10.1109/TPAMI.2007.70753

[6] Azeem, S.A., Ahmed, H. (2011). Recognition of segmented online Arabic handwritten characters of the ADAB database. In 2011 10th International Conference on Machine Learning and Applications and Workshops, Honolulu, HI, USA, pp. 204-207. https://doi.org/10.1109/ICMLA.2011.120

[7] Addakiri, K., Bahaj, M. (2012). On-line handwritten Arabic character recognition using artificial neural network. International Journal of Computer Applications, 55(13): 42-46. https://doi.org/10.5120/8819-2819

[8] Coquenet, D., Chatelain, C., Paquet, T. (2022). End-to-

end handwritten paragraph text recognition using a vertical attention network. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(1): 508-524. https://doi.org/10.1109/TPAMI.2022.3144899

[9] Coquenet, D., Chatelain, C., Paquet, T. (2020). Recurrence-free unconstrained handwritten text recognition using gated fully convolutional network. In 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), Dortmund, Germany, pp. 19-24. https://doi.org/10.1109/ICFHR2020.2020.00015

[10] Coquenet, D., Chatelain, C., Paquet, T. (2023). Dan: a segmentation-free document attention network for handwritten document recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(7): 8227-8243. https://doi.org/10.1109/TPAMI.2023.3235826

[11] AlShehri, H. (2024). DeepAHR: A deep neural network approach for recognizing Arabic handwritten recognition. Neural Computing and Applications, 36: 12103-12115. https://doi.org/10.1007/s00521-024-09674-2

[12] Rabi, M., Amrouche, M. (2024). Enhancing Arabic handwritten recognition system-based CNN-BLSTM using generative adversarial networks. European Journal of Artificial Intelligence and Machine Learning, 3(1): 10-17. https://doi.org/10.24018/ejai.2024.3.1.36

[13] Momeni, S., BabaAli, B. (2024). A transformer-based approach for Arabic offline handwritten text recognition. Signal, Image and Video Processing, 18(4): 3053-3062. https://doi.org/10.1007/s11760-023-02970-9

[14] Singh, S.S., Karayev, S. (2021). Full page handwriting recognition via image to sequence extraction. In Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, pp. 55-69. https://doi.org/10.1007/978-3-030-86334-0_4

[15] Wang, T., Zhu, Y., Jin, L., Luo, C., Chen, X., Wu, Y., Wang, Q., Cai, M. (2020). Decoupled attention network for text recognition. Proceedings of the AAAI Conference on Artificial Intelligence, 34(7): 12216-12224. https://doi.org/10.1609/aaai.v34i07.6903

[16] Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F. (2023). Trocr: Transformer-based optical character recognition with pre-trained models. Proceedings of the AAAI Conference on Artificial Intelligence, 37(11): 13094-13102. https://doi.org/10.1609/aaai.v37i11.26538

[17] Yousef, M., Hussain, K.F., Mohammed, U.S. (2020). Accurate, data-efficient, unconstrained text recognition with convolutional neural networks. Pattern Recognition, 108: 107482. https://doi.org/10.1016/j.patcog.2020.107482

[18] Wigington, C., Tensmeyer, C., Davis, B., Barrett, W., Price, B., Cohen, S. (2018). Start, follow, read: End-to-end full-page handwriting recognition. In Computer Vision-ECCV 2018: 15th European Conference, Munich, Germany, pp. 367-383. Https://doi.org/10.1007/978-3-030-01231-1_23

[19] Coquenet, D., Chatelain, C., Paquet, T. (2021). Span: A simple predict & align network for handwritten paragraph recognition. In Document Analysis and Recognition – ICDAR 2021: 16th International Conference, Lausanne, Switzerland, pp. 70–84. https://doi.org/10.1007/978-3-030-86334-0_5

[20] Chowdhury, A., Vig, L. (2018). An efficient end-to-end neural model for handwritten text recognition. arXiv preprint arXiv:1807.07965. Https://doi.org/10.48550/arXiv.1807.07965

[21] Chaudhary, K., Bali, R. (2021). EASTER: Simplifying Text Recognition using only 1D Convolutions. Canadian Artificial Intelligence Association. https://doi.org/10.21428/594757db.65eda33c

[22] Nguyen, D., Tran, N., Le, H. (2019). Improving long handwritten text line recognition with convolutional multi-way associative memory. arXiv preprint arXiv:1911.01577. Https://doi.org/10.48550/arXiv.1911.01577

[23] Chung, J., Delteil, T. (2019). A computationally efficient pipeline approach to full page offline handwritten text recognition. In 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), Sydney, NSW, Australia, pp. 35-40. Https://doi.org/10.1109/ICDARW.2019.40078

[24] Kang, L., Riba, P., Rusiñol, M., Fornés, A., Villegas, M. (2022). Pay attention to what you read: Non-recurrent handwritten text-line recognition. Pattern Recognition, 129: 108766. Https://doi.org/10.1016/j.patcog.2022.108766

[25] Such, F.P., Peri, D., Brockler, F., Paul, H., Ptucha, R. (2018). Fully convolutional networks for handwriting recognition. In 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, USA, pp. 86-91. https://doi.org/10.1109/ICFHR-2018.2018.00024

[26] Kass, D., Vats, E. (2022). AttentionHTR: Handwritten text recognition based on attention encoder-decoder networks. In Document Analysis Systems: 15th IAPR International Workshop, DAS 2022, La Rochelle, France, pp. 507-522. https://doi.org/10.1007/978-3-031-06555-2_34

[27] Wick, C., Zöllner, J., Grüning, T. (2022). Rescoring sequence-to-sequence models for text line recognition with CTC-prefixes. In In Document Analysis Systems: 15th IAPR International Workshop, DAS 2022, La Rochelle, France, pp. 260-274. https://doi.org/10.1007/978-3-031-06555-2_18

[28] Bhunia, A.K., Sain, A., Chowdhury, P.N., Song, Y.Z. (2021). Text is text, no matter what: Unifying text recognition using knowledge distillation. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, Canada, pp. 963-972. https://doi.org/10.1109/ICCV48922.2021.00102

[29] Bluche, T., Louradour, J., Messina, R. (2017). Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, pp. 1050-1055 https://doi.org/10.1109/ICDAR.2017.174

[30] Bluche, T. (2016). Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. arXiv preprint arXiv:1604.08352. https://doi.org/10.48550/arXiv.1604.08352

[31] Graves, A., Schmidhuber, J. (2008). Offline handwriting recognition with multidimensional recurrent neural networks. In Proceedings of the 21st International Conference on Neural Information Processing Systems, Columbia, Canadapp, pp. 545-552.

[32] Mustafa, W.A., Kader, M.M.M.A. (2018). Binarization

of document images: a comprehensive review. Journal of Physics: Conference Series, 1019: 012023. https://doi.org/10.1088/1742-6596/1019/1/012023

[33] Chaudhary, K., Bali, R. (2022). Easter2. 0: Improving convolutional models for handwritten text recognition. arXiv preprint arXiv:2205.14879. https://doi.org/10.48550/arXiv.2205.14879

[34] Long, J., Shelhamer, E., Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, pp. 3431-3440. https://doi.org/10.1109/CVPR.2015.7298965

[35] Badrinarayanan, V., Kendall, A., Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12): 2481-2495. https://doi.org/10.1109/TPAMI.2016.2644615

[36] Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Computer Vision-ECCV 2018: 15th European Conference, Munich, Germany, pp. 801-818. https://doi.org/10.1007/978-3-030-01234-2_49

[37] Mahmoud, S.A., Ahmad, I., Al-Khatib, W.G., Alshayeb, M., Parvez, M. T., Märgner, V., Fink, G. A. (2014). KHATT: An open Arabic offline handwritten text database. Pattern Recognition, 47(3): 1096-1112. https://doi.org/10.1016/j.patcog.2013.08.009

[38] Lopes, A., Prakash, K.B. (2023). Artificial intelligence and machine learning approaches to document digitization in the banking industry: An analysis. Ingénierie des Systèmes d’Information, 28(5): 1325-1334. https://doi.org/10.18280/isi.280521