2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-1243-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/MIUCC52538.2021.9447608

OCFormer: A Transformer-Based Model For Arabic Handwritten Text Recognition

*Aly Mostafa
Department Of Computer Science
Helwan University
Cairo, Egypt
alymostafa@fci.helwan.edu.eg

*Omar Mohamed
Department Of Computer Science
Helwan University
Cairo, Egypt
omar20170353@fci.helwan.edu.eg

*Ali Ashraf
Department Of Computer Science
Helwan University
Cairo, Egypt
aliashraf@fci.helwan.edu.eg

*Ahmed Elbehery
Department Of Computer Science
Helwan University
Cairo, Egypt
ahmedismail@fci.helwan.edu.eg

*Salma Jamal
Department Of Computer Science
Helwan University
Cairo, Egypt
Salma_20170710@fci.helwan.edu.eg

Ghada Khoriba
Department Of Computer Science
Helwan University
Cairo, Egypt
ghada_khoriba@fci.helwan.edu.eg

Amr S. Ghoneim
Department Of Computer Science
Helwan University
Cairo, Egypt
amr.ghoneim@fci.helwan.edu.com

Abstract—**The Optical Character Recognition (OCR) of Arabic historical documents is a challenging task. The reason being the complexity of the layout and the highly variant typography. Nonetheless, in recent years, with the rise of Deep learning, significant progress has been made in historical OCR; in both layout recognition and segmentation, and also in character recognition. The only downside is the limited advancements dedicated to the Arabic language, notably the handwritten text. In this paper, we present an OCR approach that utilizes state-of-the-art Deep learning techniques for the Arabic language. We built a custom dataset of obfuscated and noisy images to imitate the noise in historical Arabic documents, with a collection of 30 million images paired with their ground truth. The model utilizes both page segmentation and line segmentation techniques to enhance the resultant transcription. The model is complex enough for transcribing handwritten manuscripts. In addition, the model can detect and transcribe documents that contain Arabic diacritics. The model attained a CER of 0.0727, a WER of 0.0829, and a SER of 0.10.**

Index Terms—**OCR, Segmentation, handwritten text**

I. Introduction

While OCR technology has advanced in recent years, it lacked the accuracy needed to prove useful for historical Arabic manuscripts [8] [12] [18]. The reason for that is the frequently intricate layout, containing images, artistic border elements and ornaments, and marginal notes. Consequently, the simultaneous segmentation of text and non-text cannot be fully automated with a high degree of accuracy. The non-standardized typography also represents a challenge for the various OCR approaches. The lack of old computerized Arabic fonts prevents the easy construction of such poly-font or mixed-font models for old manuscripts; hence training individual models is essential. For a successful supervised training process, the training examples and ground truth (GT), such as line images and their corresponding transcriptions, must be prepared manually. Furthermore, Automatic lexical correction is severely hampered by extremely variable historical spelling, including regular manipulation of abbreviations. Identical terms may be pronounced differently in different books from the same age, as well as within the same text.
Recently, important milestones were the introduction of Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) [10] [6] trained using a Connectionist Temporal Classification (CTC) decoder tailored for OCRs [7], Attention mechanisms; namely, Self-Attention, Transformers [24], and Linformer (that is, Self-Attention with Linear Complexity) [25]. These methods have undeniably improved the recognition accuracy of characters and words/sentences. In this paper, we investigate how Transformers can be employed to build an OCR for historical Arabic manuscripts that is capable of (1) appropriate line and character segmentation, (2) proper transcription of handwritten manuscripts, and (3) the recognition of Arabic diacritics.

*Equal contribution

978-1-6654-1243-8/21/$31.00 ©2021 IEEE

Also, we will present a custom dataset created to combat the lack of ground truth for many historical Arabic manuscripts.

II. Related Works

Because of the diversity of handwritten character shapes and types, algorithms designed to recognise handwritten characters have had less success than those designed to recognise printed characters. Arabic character recognition is a significant task, and it is an important step in the much more difficult problems of Arabic word and sentence recognition.

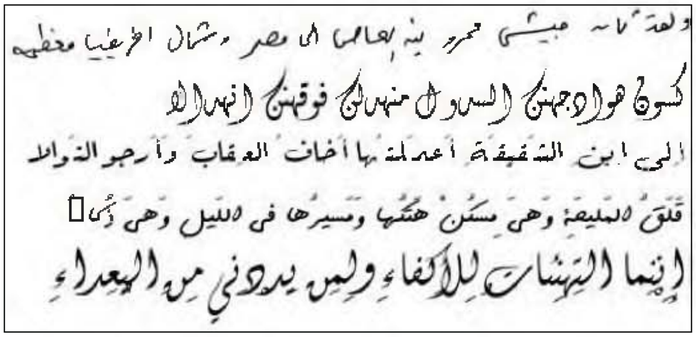
Tesseract's OCR[1]training and recognition capabilities have improved considerably (due to the addition of a new OCR engine based on LSTM [10], and Bi-LSTM [11] neural networks), empirically outperforming the character-based approaches. Tesseract excels at completely automated out-of-the-box processing of contemporary texts, but it falls short when dealing with historical documents.

OCR4all [21] uses the Calamari engine [26]; which is an OCR that uses state-of-the-art Deep Neural Networks (akin to Tesseract) and native support for pretraining and voting procedures. Graves et al'sÂ Connectionist Temporal Classification (CTC) [7] algorithm is used to train the customizable network architecture, which is composed of Convolutional Neural Networks (CNNs) and LSTM layers. OCR4all achieved a mean Character Error Rate (CER) of 0.47, tested on 17 historical English and German manuscripts. While the Calamari's architecture is robust, it only supports the English and German languages.

Ahmad et al. [1] experimented on the APTI [23] dataset using an adaptive window for feature extraction of both characters and words (achieving an Error Rate of 0.57% for the characters level, and 2.12% for words). Furthermore, they experimented using the KHATT database [16], attaining the highest CER of 1.04% for the character recognition. They also employed LSTM and its variants Bi-LSTM and MDLSTM [2] on the KPTI1 data set [15], evaluating using both normalized and non-normalized data (the CER using MDLSTM was 9.22%).

Using the KHATT dataset, Mahmoud et al [15]. developed a Hidden Markov Model (HMM) recognition system. They used text-line image pixel density, horizontal vertical edge derivatives, statistical features, and gradient features. They achieved a 51.2 % for character recognition using gradient features.

Elleuch et al. [4] presented a model that employs CNNs along with SVMs for Arabic handwriting recognition. To measure the model's performance, they used both the HACDB [14] and the IFN/ENIT [19] datasets. The authors investigated the CNN-based SVM model's performance (for Arabic characters recognition) with and without dropout. The Error rate was 5.83% and 7.05% using the HACDB and ENIT datasets respectively.



ولقد [illegible] حبشي عمرو بنه بعاصا الى مصر وشمال افريقيا مخطيه

كسون هوادجهن السدول منهن لى فوقهن انهلالا

إلى ابن الشقيقة أعملتها أخاف العقاب وأرجو النوالا

قَلَقُ المَلِيحَةِ وَهيَ مِسكٌ هَتكُها وَمَسيرُها في اللَيلِ وَهيَ ذُكا□

إِنما التهنئات لِلأكفاءِ ولِمن يدني مِن البعداءِ

Fig. 1. Sample lines of proposed Dataset

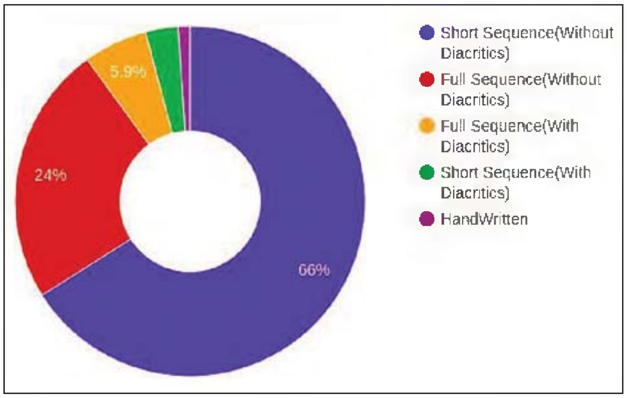


Fig. 2. The Percentages Of The Dataset

Using the APTID-MF dataset, Aziz et al [20]. implemented a segmentation-based, omni-font, For printed Arabic text, open-vocabulary OCR was used, with an average accuracy of 95%. The recognition stage has an overall accuracy of 99.97% without using font-type or other post-processing techniques.

Ahmed et al. [22] proposed a machine learning model that selects the best features for Arabic handwritten character recognition by combining neighbourhood rough sets with a binary whale optimization algorithm. Their Linear Discriminant Analysis (LDA) model achieved a recognition accuracy of 96%. using the CENPARMI dataset.

III. Dataset

The constructed dataset consists of images containing Arabic text collected from the web [28] along with their ground truth. A portion of the text includes Arabic diacritics. Multiple Arabic fonts that closely resemble the old fonts used in historical manuscripts (dating back to the 18th century) are used. There are four categories of images: Full sequences (i.e., images with more than five words), Short sequences (i.e., images that have five or fewer words), Full sequences

[1] https://github.com/tesseract-ocr/tesseract

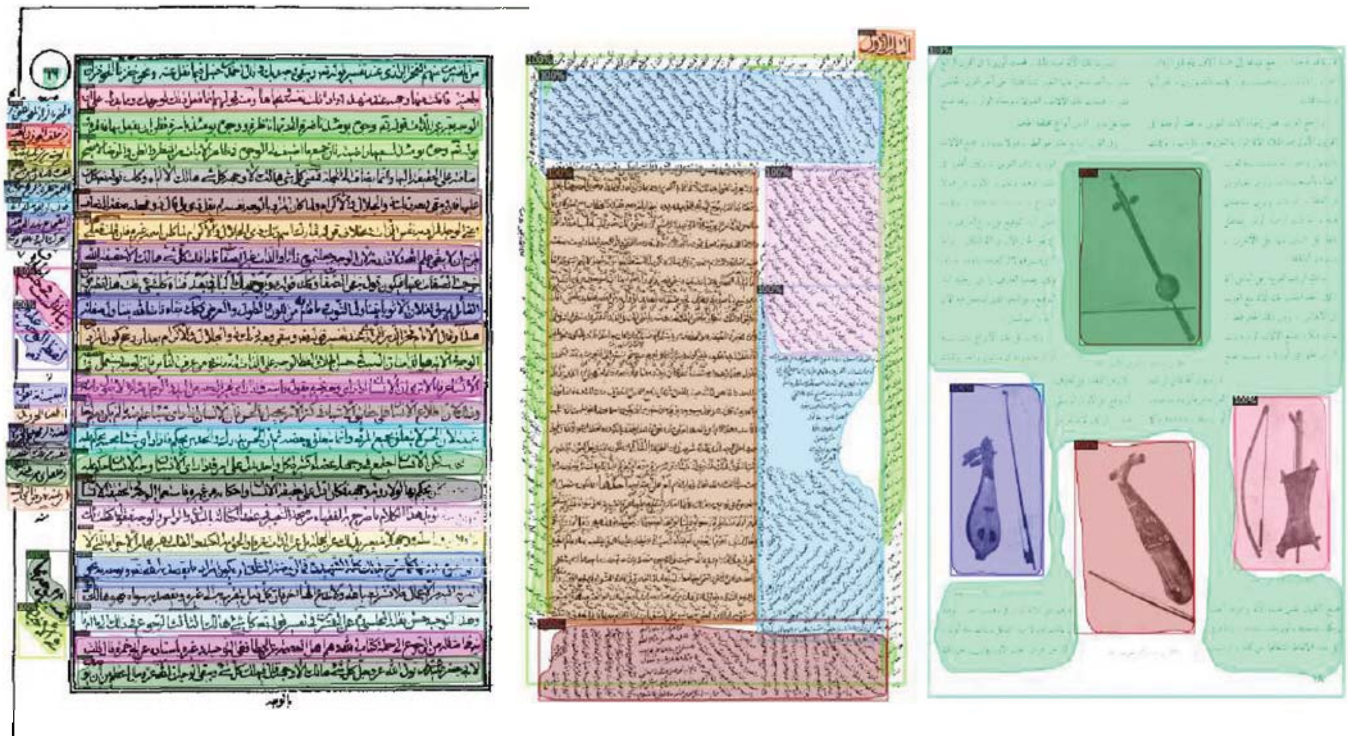


Fig. 3. Example Segmented images of historical printings

with diacritics (the images with more than five words with diacritics), and finally Short sequences with diacritics (images with five or fewer words, with diacritics).

The handwritten manuscripts from the KHATT database [16], is also included (KHATT contains unconstrained handwritten Arabic Texts written by 1000 different writers). The rationale for this is to train the model using all types of sequences and sentence positions that may appear in historical manuscripts. For example, marginal notes are Short sequences while artistic borders are Full sequences, thus ensuring that the model will be trained using all possible types of texts, and that will help in the segmentation process of handwritten manuscripts. Figure.2 illustrates the various categories of the dataset.

IV. Proposed Methodology

A. *Image Preprocessing*

1) Image augmentation: a well-known type of data augmentation that increases the images available for the training model [5], without collecting any new data. The augmentation can be done with cropping, padding, horizontal flipping, zooming, and rotating the image, while injecting as much noise as possible to ensure the model's robustness. Shearing and Changing the Luminosity are two commonly applied augmentation approaches when training large neural networks. Random Angle Rotation and Line Stretching have been applied to simulate the writing layout patterns in most historical manuscripts. For instance, marginal notes are usually written sideways in white spaces on a page.

2) Image Segmentation: Following that, the preprocessed images are segmented into regions. After removing the text-containing areas, two types of line segmentation are used. procedures are applied. Initially, layout segmentation is applied, where the model segments areas containing valid text from a page. Followed by line segmentation, where the model segments the output from the layout segmentation (i.e., text areas) into segments each corresponding to sentence (i.e., line) detected within the image. We trained a pretrained Mask-RCNN [27] on a small subset of the hand-annotated images. A pretrained model has been selected due to the convenience it provides; as it decreases the training time and does not require the manual annotation of numerous images. Figure 3 presents results of the segmentation phase (which achieved a total loss of 0.08).

B. *OCR Model*

In the context of Deep Learning (DL) techniques for image analysis, CNNs are of main importance. Therefore, we trained a CNN model (Resnet101) [9] with 101 layers as a feature extractor, resulting in the feature vectors that are crucial for the subsequent Transformer Encoder phase.

Transformers [24] are a form of neural network architecture that gained popularity after the Self-Attention mechanism was introduced (i.e., focusing on a subset of the information they get). An RNN, for example, can monitor the performance of another RNN. It focuses on different positions of that other RNN at each time stage. As a result, a

TABLE I
SELECTED ARABIC TEXT DATASETS.
(N/A) INDICATES THAT INFORMATION IS NOT AVAILABLE

Dataset	Number of Words	Number of characters	Number of Fonts	Number of Font Size	Number of Font Styles
APTI [23]	113,284	648,280	10	10	4
IFN/ENIT [19]	26459	212,211	1 (Handwritten)	N/A	1
HACDB [14]	N/A	6600	1 (Handwritten)	N/A	1
APTID / MF [19]	N/A	27,402	10	4	10
KHATT [15]	400	7900	1 (Handwritten)	N/A	1
Proposed Dataset	270m	1.6 billion	13	13	13

Transformer is a model that uses Self-Attention to increase the speed of training in Deep Learning models. Transformers also solve the parallelization problem. The proposed OCR Model utilizes CNNs together with a Transformer. Transformers consist of two modules, an encoder, and a decoder, which have customizable parameters that can affect a model's performance based on its task. Our purposed model is initialized with four encoders, four decoders, four attention heads, and 256 hidden dimensions.

The **encoder** module receives the features extracted by the CNN as an input, then performs embedding and positional encoding, creating a sequence input that can be passed to the Multi-Head Attention. The Multi-Head Attention layer performs Self-Attention, then adds a layer-normalization before passing the feature vector (of 2048 elements) to a feed-forward network that applies layer-normalization on residuals. These operations are performed four times (because the model is initialized with four encoders).

Regarding the **decoder** module, its input is the text to be extracted from the images, which will undergo similar operations to that of the encoder. The embedding layer performs embedding and positional encoding, creating the sequence that will be passed to the masked Multi-Head Attention layer. The masked Multi-Head Attention layer performs Self-Attention on the text sequence, then adds layer-normalization before performing cross attention between the encoder's output and the text sequence received from the Multi-Head Attention of the decoder. Then it applies layer- normalization and passes the vector to a feed-forward network and a linear layer. Finally, a Softmax function produces the probabilities for the predicted words.

V. RESULTS

This section presents the results of the conducted experiments for training the proposed OCR model on the constructed dataset. Due to the lack of resources and computational power, all the experiments were conducted using a subset of 15000 single-line images (instead of the full dataset of 30.5 million single-line images), and the model was tested using 5 randomly selected testing subsets (of 2000 single-line images each). In the first set of experiments, the OCR model was trained using only a single font, without diacritics, and with both long and short sequences, achieving a CER of 0.0032, a Word Error Rate (WER) of 0.0058, and a Sequence Error Rate (SER) of 0.088. Subsequently, the OCR model was trained using all 12 fonts available, with diacritics, and with both long and short sequences, achieving a CER of 0.0727, a WER of 0.0829, and a SER of 0.10.

VI. DISCUSSION

A. *Proposed Dataset*

The proposed dataset is a large, multi-font, and multi-style, text recognition dataset in Arabic. Various considerations must be met while constructing the dataset to ensure the diversity of the writing styles. That includes different fonts, different styles, and different noise patterns on the characters used while generating the images. The database is constructed using 13 Arabic fonts and multiple font styles. The dataset contains 30.5 million single-line images, including more than 270 million words and 1.6 billion characters. The ground truth, style, and font used to generate each image are available.

The generated text images vary according to the following:

1. Twelve Different Fonts: Alsamt Diwani, Barada Reqa, Diwani Letter, HSN Naskh Farisi, M Unicode Abeer, Mj Ghalam, Nawel, Old Antic Outline, Phalls Khodkar, Sahel, Tarwat Emara Ruqaa Hollow, and Tarwat Emara Ruqaa Light.

2.Thirteen Different Sizes; one for each font.

3. Multiple Styles: Bold, Plain, Italic, Italic and Bold, etc.

4. Various Forms of Overlap among characters due to the different fonts and large combinations of words and characters.

5. Enormous Vocabulary, which allows the models to be tested on novel unseen data.

6. Multiple downsampling and antialiasing filter artefacts caused by the random addition of white-pixel columns at the beginning and end of lines in the images.

7. Each image's height and width are variable.

The last point in a word is crucial to the series of characters that appear in it. There is no prior knowledge of the text's location in the picture, so the baseline must be calculated by the recognition model.

B. *Proposed Model*

In this section, we will emphasize some details of the model. According to Vaswani et al. [24] and Devlin et al [3], the model's performance to generalize to unseen data increases proportionally with the number of parameters, which are the (1) hidden dimensions, (2) number of encoder layers, (3) number of decoder layers, and (4) number of heads. The base transformers' model of Vaswani et al. [24] achieved a BLEU of 38.1 (in the English-to-French), while the big transformers' model achieved a BLEU of 41.8, and the same follows with the BERT model [3]. We employed the Adam optimizer [13] with weight decay, and label smoothing [17]. The learning rate was initialized to 0.0001, then an adaptive learning rate technique was applied (as follows: if the validation loss increases for 3 consecutive iterations, the learning rate changes according to the LR scheduler). The model was trained for 700 update steps. In Sequence-to-Sequence (seq2seq) models, The most widely used decoding algorithms are greedy search and beam search [29]. At each time level, the greedy search seeks out the token with the highest conditional probability. Beam search, on the other hand, is a better version of greedy search that, unlike greedy search, selects the k highest conditional probabilities at each time level. With each passing time phase, Based on the k tokens from the previous time stage, it chooses the k highest conditional probabilities.

VII. Future Works

In this research, we proposed and investigated a novel approach for harnessing the power of Transformers to transcribe historical Arabic manuscripts. It is expected that this approach will help speed up, and increase the accuracy of, the OCR process. In our future work, we aim to improve the model's accuracy by training the model with the entire constructed dataset (allowing the model to train on all the data will significantly increase the accuracy of the OCR). As well, we aim to increase the number of encoders, decoders, and other parameters that can highly affect the model's performance and accuracy. Finally, we aim to optimize the model's time and space complexity from quadratic complexity to linear complexity.

References

[1] Irfan Ahmad, Sabri A Mahmoud, and Gernot A Fink. Open-vocabulary recognition of machine-printed arabic text using hidden markov models. *Pattern recognition*, 51:97–111, 2016.

[2] Théodore Bluche, Jérôome Louradour, and Ronaldo Messina. 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)*, volume 1, pages 1050–1055. IEEE, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Mohamed Elleuch, Najiba Tagougui, and Monji Kherallah. Optimization of dbn using regularization methods applied for recognizing arabic handwritten script. *Procedia Computer Science*, 108:2292–2297, 2017.

[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[6] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[7] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[8] Tobias Gruuening, Gundram Leifert, Tobias Strauss, and Roger Labahn. A robust and binarization-free approach for text line detection in historical documents. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 236–241. IEEE, 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[12] Philip Kahle, Sebastian Colutto, Günter Hackl, and Günter Mühlberger. Transkribus-a service platform for transcription, recognition and retrieval of historical documents. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 4, pages 19–24. IEEE, 2017.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Ahmed Lawgali, Maia Angelova, and Ahmed Bouridane. Hacdb: Handwritten arabic characters database for automatic character recognition. In *European Workshop on Visual Information Processing (EUVIP)*, pages 255–259. IEEE, 2013.

[15] Sabri A Mahmoud, Irfan Ahmad, Wasfi G Al-Khatib, Mohammad Alshayeb, Mohammad Tanvir Parvez, Volker Märgner, and Gernot A Fink. Khatt: An open arabic offline handwritten text database. *Pattern Recognition*, 47(3):1096–1112, 2014.

[16] Sabri A Mahmoud, Irfan Ahmad, Mohammad Alshayeb, Wasfi G Al-Khatib, Mohammad Tanvir Parvez, Gernot A Fink, Volker Märgner, and Haikal El Abed. Khatt: Arabic offline handwritten text database. In *2012 International Conference on Frontiers in Handwriting Recognition*, pages 449–454. IEEE, 2012.

[17] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019.

[18] Clemens Neudecker, Konstantin Baierer, Maria Federbusch, Matthias Boenig, Kay-Michael Würzner, Volker Hartmann, and Elisa Herrmann. Ocr-d: An end-to-end open source ocr framework for historical printed documents. In *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage*, pages 53–58, 2019.

[19] Mario Pechwitz, S Snoussi Maddouri, Volker Märgner, Noureddine Ellouze, Hamid Amiri, et al. Ifn/enit-database of handwritten arabic words. In *Proc. of CIFED*, volume 2, pages 127–136. Citeseer, 2002.

[20] Aziz Qaroush, Abdalkarim Awad, Mohammad Modallal, and Malik Ziq. Segmentation-based, omnifont printed arabic character recognition without font identification. *Journal of King Saud University-Computer and Information Sciences*, 2020.

[21] Christian Reul, Dennis Christ, Alexander Hartelt, Nico Balbach, Maximilian Wehner, Uwe Springmann, Christoph Wick, Christine Grundig, Andreas Büttner, and Frank Puppe. Ocr4allâan open-source tool providing a (semi-) automatic ocr workflow for historical printings. *Applied Sciences*, 9(22):4853, 2019.

[22] Ahmed Talat Sahlol, Mohamed Abd Elaziz, Mohammed AA Al-Qaness, and Sunghwan Kim. Handwritten arabic optical character recognition approach based on hybrid whale optimization algorithm with neighborhood rough set. *IEEE Access*, 8:23011–23021, 2020.

[23] Fouad Slimane, Rolf Ingold, Slim Kanoun, Adel M Alimi, and Jean Hennebert. Database and evaluation protocols for arabic printed text recognition. *DIUF-University of Fribourg-Switzerland*, 2009.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[25] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[26] Christoph Wick, Christian Reul, and Frank Puppe. Calamari-a high-performance tensorflow-based deep learning package for optical character recognition. *arXiv preprint arXiv:1807.02004*, 2018.

[27] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[28] Waleed A. Yousef, Omar M. Ibrahime, Taha M. Madbouly, Moustafa A. Mahmoud, Ali H. El-Kassas, Ali O. Hassan, and Abdallah R. Albohy. Poem comprehensive dataset (pcd), 2018.

[29] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. 2020. https://d2l.ai.