Journal of Theoretical and Applied Information Technology

31st December 2023. Vol.101. No 24
© 2023 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TRANSFORMER-BASED MODEL FOR HANDWRITTEN RECOGNITION ARABIC WORDS AL-SOUDANI MAGHREBI SCRIPT

SIDI AHMED MAOULOUD[1], M. H OULD MOHAMED DYLA[2], CHEIKH BA[3]

[1,3]University Gaston Berger of Saint-Louis, UFR Applied Sciences and Technologies, Senegal

[2]University of Nouakchott, Modeling And Scientific Computation, Mauritania

E-mail: [1]maouloud.sidi@ugb.edu.sn_, [2]mohdyla@gmail.com, [3]cheikh2.ba@ugb.edu.sn

ABSTRACT

Automatic handwriting recognition is a crucial element for various applications across different domains. It's a complex problem that has garnered significant attention over the past three decades. For the Arabic language, research has primarily focused on recognizing historical Eastern Arabic scripts and manuscripts. However, fewer studies have been conducted on Maghrebi Arabic scripts and their variants. In this article, we introduce a novel dataset of Arabic words written in the widely used Maghrebi Al-Soudani script, primarily found in West Africa, extracted from three different manuscripts. Our dataset comprises 30,430 words. We also propose a Handwritten Text Recognition (HTR) model based on an Encoder-Decoder architecture, utilizing a Convolutional Neural Network (CNN) for image encoding and a Transformer for image-to-text decoding. We train our model, as well as a recent reference model (FPHR), on this dataset. The results demonstrate promising performance of our model, outperforming the FPHR model with accuracy rates of 10% for Character Error Rate (CER) and 9.9% for Word Error Rate (WER).

Keywords: *Dataset Transformer Encoder-Decoder Handwritten recognition Al-Soudani Arabic script Manuscript Image-to-Sequence*

1. INTRODUCTION

Arabic is a cursive language with different character shapes and words composed of one or more subwords. In addition, Arabic writing can be subject to character contact and overlap, which complicates the recognition of touching or overlapping characters, and limits the results of OCR models, as two overlapping or touching characters may be mistakenly considered as one [2], [5]. For these reasons, [1] it has been found that Arabic OCR programs perform notoriously poorly, despite the optimistic claims of some of their marketing materials. Mansoor Alghamin their published article 2017 [1] assert that "although handwriting is significantly more difficult than printed Arabic text for OCR, OCR of printed Arabic text still poses significant challenges.". After evaluating Sakhr, Finereader, RDI Clever Page, and Tesseract (version 3), the main options for OCR on Arabic print, they conclude that "all evaluated Arabic OCR systems have low accuracy rates, below 75%" [16].

Printed Latin text recognition is an area in which deep learning has proven itself perfectly. However, the situation

of Latin script (HTR) text recognition, especially historical documents are less satisfactory [3], [4]. Since there are only a few open datasets, the quality of trained pattern recognition is low.

Despite the fact that the processing of Arabic historical documents is a recent research topic, it has experienced remarkable growth in recent years [18]. This growth is due to the emergence of some specialized dataset of annotated Arabic historical documents. While the processing of Arabic historical documents remains a particularly difficult problem. First, because of the complicated nature of the Arabic script compared to other scripts and second, because the documents are old [18], [9].

However, the specialized databases cited above often focus on oriental historical Arabic documents and writings, leaving out many poorly documented Arabic written traditions.

In this paper, we present a new architecture model based on an encoder-decoder usinng covolution (CNN) for encoding the image, and a

Transformer to decoding the image into text. We are also proposing a new dataset for the recognition of handwritten Arabic words. This is the first dataset focusing on the Arabic script called Al-Soudani script which was a widely used script in Islamic sub-Saharan Africa until the 19th century. This Al-Soudani script is used in many manuscript collections and libraries in West Aftica such as Ahmed Baba Centre in Timbuctu.

The Maghrebi script reached all the centers of sub-Saharan Africa, and the Muslims wrote their writing manuscripts using this script, and even wrote their local languages in Arabic script as well, where they wrote the language of Borno, Fulani, Hausa, Wolof and Singay. These languages borrowed many Arabic words and expressions, wich became part of these local languages, and the local African qualities were embodied in Maghrebi calligraphy in a way that gave it a diversity and a unique African tinge [20]. This African Arabic calligraphy is known as the Al-Soudani script (see Figure 1).

Al-Soudani script characters are characterized by being simple, thick and dry, reflecting the simplicity and harshness of life in the African desert. However, it retains the outstanding features of the Maghrebi calligraphy, especially the Al-Mebsout one, the lines are alternately thick and thin. The vertical bars rise to a great height, out of proportion with the size of the writing and the shape of the loops. The general slope of the writing is strongly accentuated and directed towards the left [22]. The Al-Soudani script is a very particular variant of the Maghrebi script. It has specific characteristics and presents many new challenges for HTR architectures. Table I presents some of these features.

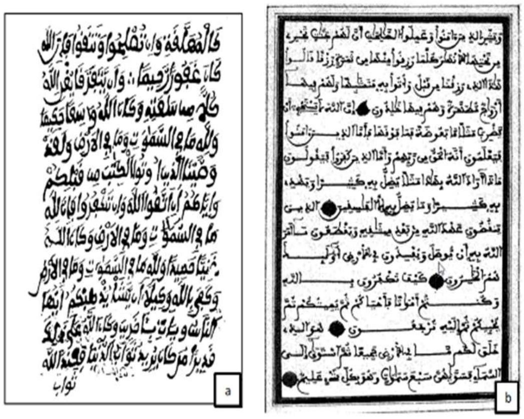


Fig 1: (a)Al-Soudani script with the strokes of (b)Al-Mebssout calligraphy

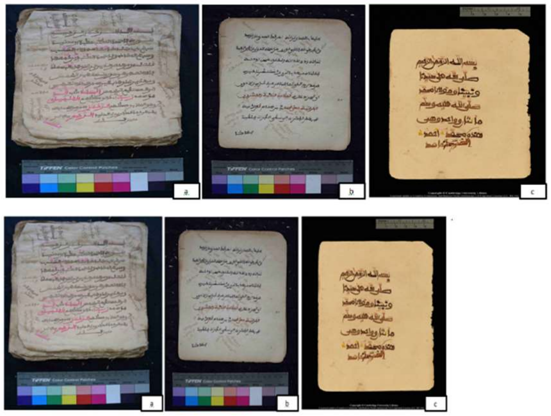


Fig2;(a)EAP1042_Ibrahima_Sagna_M005_pt1(p.1) ,(b)EAP1042_Abdou_Solly_M001_pt2(p.1) ,(c)MS Or.2251(p. 1)

2. RELATED WORKS

To evaluate algorithms for the analysis and recognition of historical Arabic documents, researchers need datasets. In the literature, only a few datasets of historical Arabic documents exist such as: VML-HD Dataset a dataset [6], published in 2017 for word spotting and word recognition tasks. The dataset is composed of five manuscripts; each manuscript had been written by one distinct scribe, from the year 1088 to 1451. The HADARA80P [7] dataset was proposed in 2014 by a German institute. It is composed of 80 pages of the historical book ”Taaun”. These pages correspond to the book cover and the first 79 first pages. The HADARA80P dataset contains 16,720 annotated words. Likewise the IBN-SINA Dataset, [10] was proposed in 2010, by a Canadian University and provided by the Institute of Islamic Studies (IIS). The dataset contains 51 folio of one manuscript, the kšf altmwyhāt fy šrḥ ālišārāt w āltnbyhāt.

The public BADAM dataset[8] contains 42 manuscripts from four digital collections of the Arabic and Persian languages. It contains 400 annotated pages from different domains and time periods. The MHDID Dataset [11] (Multi-distortion Historical Document Image Database) contains 335 historical document images with size of 1024x1280 pixels The VDIQA Dataset [12] (Visual Document Image quality assessment), contains 177 histhorical document images with of 1024x1280 pixels.The dataset was collected from the library of Qatar University and documents were edited between the 1st and the 14th islamic centuries period, and KERTAS [13], dedicated to manuscripts datation.

Recently a dataset called RASAM [9] for the recognition and analysis of Maghrebi Arabic script. RASAM containing 300 images representative of the Maghrebi Arabic script production in North Africa. RASAM does not distinguish between the main variants of the Maghrebin script widely used in this area, namely Al-Mebssout, Al-Moujawhar, Al-Moussnad, each of which has its own characteristics. RASAM does not contain any images of Al-Soudani script.

In the literature on historical Arabic handwriting recognition, most proposed models are either a CNN [34] [35] or a combination of CNN with Recurrent Neural Networks (RNN) [36], widely used for sequence modeling in historical HTR (Handwritten Text Recognition). However, RNN variants face issues of gradient vanishing or explosion, where models fail to learn information across long sequences. The Transformer model with an attention mechanism has been introduced, demonstrating superior performance [26] compared to RNNs and the combination of RNN with CNN.

In this paper, we propose a new Encoder-Dncoder model whose decoder is a Transforemr. Our model is an improvement on the previous model [26]. As a result, our model has shown better results than [26] in the recognition of words in Soudani Maghribi Arabic script

3. DATASET

A. Sources Dataset in script Al-Soudani Arabic Maghrebi

The main source of the dataset consists of two manuscripts from the Boston University Libraries that are written in Al-Soudani scripts. (i)AnArabicmanuscript(EAP1042_Ibrahima_Sagna_M005.pt1)[21], which is a copy of the first part of the tfsyr ālğlālyn, a classical exegesis of the Qur'an widely used in Sunni Muslim communities around the world. It was produced by a teacher, named Jalal al-Din al-Mahali(date of death 1459) and his student,Jalāl al-Din Abu al-Fadl Abd al-Rahman ibn Abi Baker al-Suyūtī (died 1505). (ii)The second manuscript is a copy of mqāmāt ālḥryry (EAP1042_Abdou-_Solly_M001_pt2) [21], with numerous glosses in Arabic and Soninke ajami[1]. The work is widely known among Arab and Muslim scholars. It was written by Abū Muhammad al-Qāsim ibn Muhammed ibn Uthmān al-Harīrī, also known as al-Harīrī of Basra (1054-1122). It was copied by Arfang Fanding Solly, the grandfather of the current owner of the manuscript, (iii) a third manuscript(MS Or.2251) from the University of Cambridige Digital Library[2] written in Al-Sudan script is a Late African copy of thedlāyl ālhyrāt, a 'manual' consisting of blessings and prayers for daily life and in particular for pilgramige to Mecca. Partly composed of selections from the Qur'an and sayings of the prophet, the original work is attributed to the Sufi Abū Abdullah Muhammad ibnSulaymān ibn Abū Baker Activer Windows al-Jazouli al-Simlālī al-Hassani (date of death 1465), who lived in Marrocco This text was copied by hand throughout the Islamic world, from northwest Africa to southeast Asia, until the last century(see Figure 2). The images of the final dataset is in JPEG format and has a resolution between 100 DPI(dot per inch) and 200 DPI.

B. Segmentation Dataset

We first manually transcribed the handwritten text source images into a text file containing the ground truth text, then we segmented the handwritten text images into words using the Labelme[3] software from which each handwritten word image is associated with its truth text. A JSON file is generated for each page containing the truth text and the coordinates of the corners of the rectangle encompassing the handwritten word.

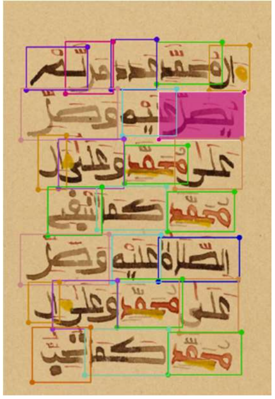


Fig 3: Example of word segmentation by rectangles in Labelme for a single

1) Words Cut:

We chose to use the rectangle (see Figure 3) segmentation instead of polygons because we want to create a dataset for handwritten word recognition in a contextual way and not isolated words as the case of the IAM [15] and most reference datasets for handwritten word recognition. In this way we will take into account vertical and horizontal overlaps between characters and especially between words. These overlaps are frequent in Arabic manuscripts and in particular in

[1] 2Ajami: which comes from the Arabic root for foreign or stranger, is an Arabic script used for writing African languages.

[2] https://cudl.lib.cam.ac.uk/view/MS-OR-02251/1

[3] https://github.com/wkentaro/labelme

Al-Soudani Arabic script manuscripts, and they are a main cause of the higher Character Error Rate (CER), and Word Error Rate (WER) of the OCR and HTR of the Arabic handwritten text [14] pink rectangle whose annotation the word يصل its last letter ل overlaps with the first letter ع of the word surrounded by the blue rectangle annotated by عليه (see Figure 4, no. a), also in ditto there is an overlap between the two words الوالي، وجعل (see Figure 4, no. b). The overlapping pixels will be learned both by the learning modules as part of the first character and part of the second character. In other words, the set of pixels in (Figure 7, no. a) compared to the Latin language. We can notice the reciprocal overlapping between the characters of the words succeeding (see Figure 3). In the image Surrounded by the will be learned as an ل or a sequence of ل, likewise the group of pixels in (Fig 7, no. b)

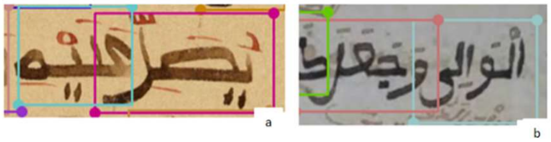
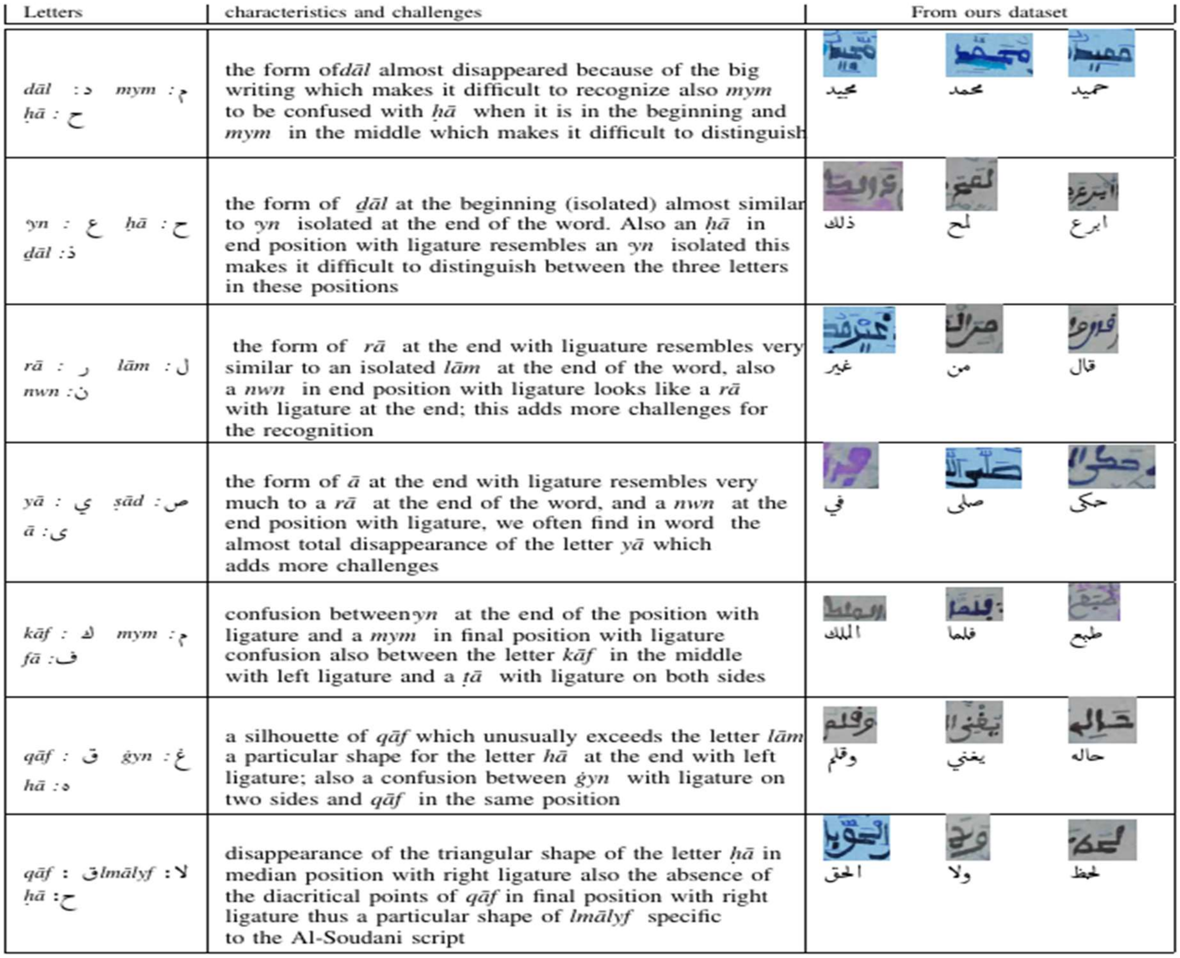


Fig 4 : Examples of words overlap

Table I: Some Characteristics and Challenges.



Letters	characteristics and challenges	From ours dataset		
dāl : د *mym* : م *ḥā* : ح	the form of*dāl* almost disappeared because of the big writing which makes it difficult to recognize also *mym* to be confused with *ḥā* when it is in the beginning and *mym* in the middle which makes it difficult to distinguish	مجيد	محمد	حميد
ʿyn : ع *ḥā* : ح *ḏāl* : ذ	the form of *ḏāl* at the beginning (isolated) almost similar to *ʿyn* isolated at the end of the word. Also an *ḥā* in end position with ligature resembles an *ʿyn* isolated this makes it difficult to distinguish between the three letters in these positions	ذلك	لح	ابرع
rā : ر *lām* : ل *nwn* : ن	the form of *rā* at the end with liguature resembles very similar to an isolated *lām* at the end of the word, also a *nwn* in end position with ligature looks like a *rā* with ligature at the end; this adds more challenges for the recognition	غير	من	قال
yā : ي *ṣād* : ص *ā* : ى	the form of *ā* at the end with ligature resembles very much to a *rā* at the end of the word, and a *nwn* at the end position with ligature, we often find in word the almost total disappearance of the letter *yā* which adds more challenges	في	صلى	حكى
kāf : ك *mym* : م *fā* : ف	confusion between*ʿyn* at the end of the position with ligature and a *mym* in final position with ligature confusion also between the letter *kāf* in the middle with left ligature and a *ṭā* with ligature on both sides	الملك	فلما	طبع
qāf : ق *ġyn* : غ *hā* : ه	a silhouette of *qāf* which unusually exceeds the letter *lām* a particular shape for the letter *hā* at the end with left ligature; also a confusion between *ġyn* with ligature on two sides and *qāf* in the same position	وقلم	يغني	حاله
qāf : ق*lmālyf* : لا *ḥā* : ح	disappearance of the triangular shape of the letter *ḥā* in median position with right ligature also the absence of the diacritical points of *qāf* in final position with right ligature thus a particular shape of *lmālyf* specific to the Al-Soudani script	الحق	ولا	لحظ

will also be learned as ع or a sequence of ع.

The JSON file generated by Labelme per page is browsed by a python script to cut each rectangle and save it in a PNG file at the same time a text file is generated containing the annotations of each rectangle concatenated with the name of the corresponding PNG file (see Figure 6). Finally the

PNG files of all the cut rectangles of the annotated pages are combined in a single directory, and all their corresponding text files containing the annotations are also, copied into a single text file, to constitute the dataset
.

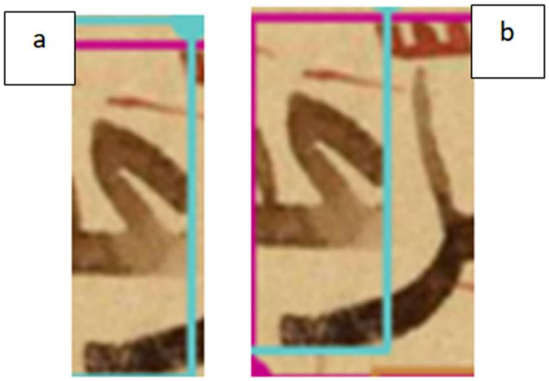
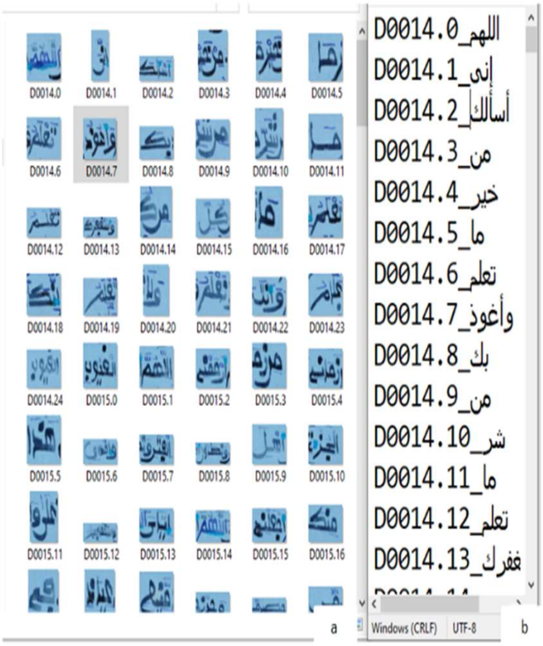


Fig 5 : Examples of characters overlap

2) Synthetic data:

To increase the size of the dataset by new samples we chose to segment into characters two new pages from two different sources which gave us 178 characters, from which we created 27234 images of Al-Soudani handwriting containing a string of characters ranging from 2 to 4 characters.



```
D0014.0_اللهم
D0014.1_إنى
D0014.2_أسألك
D0014.3_من
D0014.4_خير
D0014.5_ما
D0014.6_تعلم
D0014.7_وأغوذ
D0014.8_بك
D0014.9_من
D0014.10_شر
D0014.11_ما
D0014.12_تعلم
D0014.13_غفرك
```

Fig 6 : (a) Rectangles cut in PNG file;(b) their annotations in text file

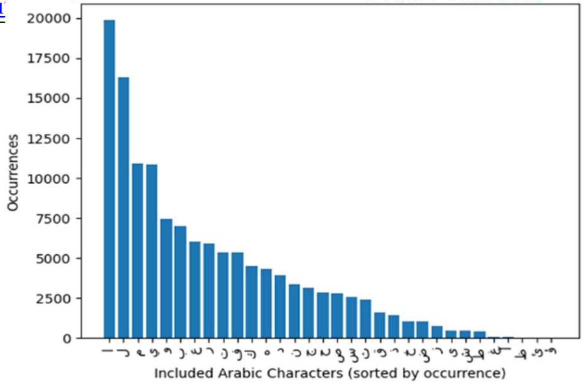


Fig 7: Histogram of Occurrences of Included Arabic Characters in ours dataset

C. *Final Dataset*

The current version(1.0) of the dataset contains 3196 images and 27234 synthetic images, which makes a total of 30430 images annotated with words handwritten in script Al-Soudani PNG format format with a resolution between 100 DPI (dot per inch) and 200 DPI and its annotations in a single text file. This dataset is divided into 80% for training and 15% for validation and 5% for testing, (see Figure 6). The datatset is designed for the recognition of Al-Soudani script words in their context and with overlap, hence an HTR model is required for training.

According to figure 7, we can conclude that the histogram is decreasing. This is because of the fact that some characters occur more frequently in the Al Maghribi Al-Suodani script than others.

4. PROPOSED MODEL

A. *Motivation*

The Sequence-to-Sequence models using RNNs (Recurrent Neural Network) with attention mechanism proposed in 2014 in particular by Yoshua Bengio in [24] have opened up a new field of research in particular in NLP (Neural language Processing) by presenting only Deep Learning models improved and surpassed the performance of older models. However, the sequential nature of RNNs was then regularly singled out as an obstacle to training these models on long texts, both for questions of time (even modern GPUs do not parallelize this type of process well) and vanishing gradient of despite the use of LSTM (Long Short Term Memory Network Models).

The Transformer model presented in June 2017 in the paper [25] has revolutionised both the encoding of a sequence but also the attention mechanism to be used. The encoding is no longer sequential but the matrix of the whole sequence and the attention mechanism called self-attention is no longer single but propagates 3 times. It allows for better performance and more parallel computations, which makes it faster to train than RNN-based models. As shown in Figure. 8 our model presented, to recognize handwritten arabic words from Al-Soudani script. We substituted the present Resenet [27] by wideResnet [28] Encoder. It is widened architecture for ResNet blocks that allows for residual networks with significantly improved performance, which contains less layers but faster than Resent. wideResnet-50-2-bottleneck outperforms ResNet152 having 3 times less layers, and being significantly faster [28]. We also tried to fine-tune the hyperparameters a bit more to try and improve performance, for the recognition of Maghribi Al-Soudani Arabic scripts. In the following sections, the encoder and the decoder are described in more detail.

1) Encoder:

The encoder uses a CNN to extract a 2D feature-map from the input $I_i \in R^{h\times w\times c}, 0 < i \leq$ card(dataset) with h, w and c being respectively the height, the width, and the number of channels (c = 3 for an RGB image), We uses the wideResnet50_2[28] architecture without its last two layers: the average pool and linear projection, and change the first layer convolution to take as input an image: $I_i \in R^{h\times w\times 1}$, converted into grayscale to lighten the calculations. It extracts some feature maps for the whole imput image: $f_{2d} \in R^{h_f\times w_f\times d_{model}}$; where d_{model} is the Transformer's hidden-size (the dimension of positional encodings and the embeddings input target to Transformers layers).

The original transformer paper [25] offers the $1D$ sequences. Since the inputs are $2D$ images, we replaced $1D$ positional encoding with $2D$ positional encoding, as proposed in [29], this $2D$ positional encoding ($P_{2d} \in R^{h_f\times w_f\times d_{model}}$;) is added to feature maps and finally flattened into a $1D$ sequence(Equation 2). The $2D$ positional encoding help Transformer decoder better identifies the image feature vector locations. $2D$ positional encoding is a fixed sinusoidal encoding as in [25], but using the first $\frac{d_{model}}{2}$ channels to encode the Y coordinate and the rest to encode the X coordinate (Equation1) .

$$P_{2d} = \begin{cases} \sin(w_k, y), \text{if } i = 2k \\ \cos(w_k, y), \text{if } i = 2k + 1 \\ sin(w_k, x), \quad \text{if } i = \frac{d_{model}}{2} + 2k \\ \cos(w_k, x), \text{if } i = \frac{d_{model}}{2} + 2k+ \end{cases} \qquad (1)$$

Where

$$w_k = {}^{1}\!/_{10000^{2k/d_{model}}} \quad k \in [0, d_{model}/4]$$

$$f_{1d} = flatten(f_{2d} + P_{2d}), f_{1d} \in R^{L_1\times d_{model}} \qquad (2)$$

Where

$$L_1 = h_f \times w_f$$

2) Transformer Decoder:

The decoder module we used follows the standard transformer architecture in [25]. Its main role is to use an autoregressive model to predict the decoded character sequence $(Seq = (s_0, \ldots, s_{t-1}))$ by assisting the visual features $f_{1d} \in R^{L_1\times d_{model}}$ generated by the encoder to predict the next character sequence, and outputs the probabilities Y_t for each token(character) in the alphabet A at time step t.

Embeddings are used to convert each of the tokens (st) into $\overrightarrow{E_{st}}$ vector of dimension d_{model}. The embeddings are summed with a $1D$ positional encoding specifying the position of the predicted token $\left(p_t^{(i)}\right)$ in the predicted sequence:

$$\overrightarrow{q_t}^i = \overrightarrow{p_t}^i + \overrightarrow{E_{s_t}}^i$$

Where $\overrightarrow{q_t} \in R^{d_{model}}$

$$\overrightarrow{p_t}^i = \begin{cases} \sin(w_k, t), \ if \ i = 2k \\ \cos(w_k, t), \ if \ i = 2k + 1 \end{cases}$$

Where

$$w_k = {}^{1}\!/_{10000^{2k/d_{model}}} \quad \overrightarrow{p_t} \in R^{d_{model}}$$

The decoder consists of a stack of transformer decoder layers (shown in Figure. 8) followed by a convolutional layer with the 1×1 kernel and a function max(softmax(.)) which calculates the next token probabilities Y_t . Transformer decoder layers are based on multi-head attention mechanisms [25], each transformer layer contains two blocks of multi-head attention:

Self-attention aims to model the dependencies between the predicted sequence: it is multi-headed attention where Q queries, keys K and V values are generated from the same input which is the vectors $\overrightarrow{q_t}^i$:

$$Attention^{(t)}(q,k,v) = softmax(\frac{qk^t}{\sqrt{d_k}})v$$

Where

$$q = \overrightarrow{w_q}\,\overrightarrow{q_t}^{(i)}$$
$$k = \overrightarrow{w_k}\,\overrightarrow{q_t}^{(i)}$$
$$v = \overrightarrow{w_v}\,\overrightarrow{q_t}^{(i)}$$

In practice, the self-attention output matrix can be computed in parallel by the query Q, the key K, the value V and the dimension d_k:

$$MultiHead(Q,K,V) = Concat(head_1, \ldots, head_h)W^0$$

Where

$$head_i = Attention(W_i^Q Q, W_i^K K; W_i^V V)$$

and the projections are parameter matrices(learned during training)

$W_i^Q \in R^{d_{model} \times d_k}, W_i^k \in R^{d_{model} \times d_k}, W_i^V \in R^{d_{model} \times d_v}, W^o \in R^{n_h \times d_{model}}$

In this model we employ $d_{model} = 260$ and $n_h = 4$ parallel attention layers, or heads. For each of these we use

$$d_K = d_v = \frac{d_{model}}{n_h} = 65.$$

Self-attention is causal since it is based on the previous predictions. In the training phase, a lower triangle mask matrix is applied to QK^t before the softmax for enable the self-attention module to restrict the attention region for each time step. Due to masked multi-head attention mechanism, the whole training process requires only one forward computation.

- Cross attention which helps the decoder focus on appropriate places in the input sequence. Cross attention is used to extract visual information from the encoder and takes as inputs the matrices:

$$K = W_j^K f_{1d}, V = KW_j^V f_{1d}.$$
$$Q = LayerNorm(MultiHead(Q,K,V) + Q_H)$$

Where

$$Q_H = [q_0^{(i)} q_1^{(i)} \ldots . q_{t-1}^{(i)}]$$

And

$$LayerNorm(q_{ij}) = \gamma \frac{q_{ij} - \mu_Q}{\sqrt{\sigma_Q^2 + \epsilon}} + \beta \; with: q_{ij} \in Q$$

The block applies, in sequence for each layer of the transformer: a self-attention layer(that we just talked about), layer normalization, a feed forward layer (a single MLP applied independently to each vector), and another layer normalization. Residual connections are added around both, before the normalization.

3) Implementation Details:

We used WideResNet CNN, at 50 layers, for the encoder part; the motivations for this choice of CNN are already cited. In the decoder part, we use the standard transformer model. We set the embedded dimension and model dimension to $d_{model} = 260$, the number of heads in the multi-head attention module to $n_h = 4$, the dimension of inner-layer in the FFN (Feed Forword Network) to $d_{ff} = 1024$, and the number of transformer decoder layer to N = 6. The 0.5 dropout rate is used to prevent overfitting, and activation function inside feed-forward layer is GELU [30](Gaussian Error Linear Unit), GELU performs better than ReLU and ELU activation functions [30].

4) Training:

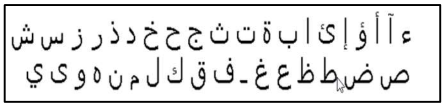
the loss function will then be the sum of the loss over each token. The loss will be cross-entropy(Equation 3), where the token is A(Our Vocabilay) in the correct label to predict. Take note, that because the sequence can be padded by token special (PAD) during training, the loss is also masked to ignore the result on padded tokens.

$$Loss = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \qquad (3)$$

where $y_{o,c}$ is a binary indicator (0 or 1) if class label c is the correct classification for token o $p_{o,c}$ is predicted probability token o is of class c, and $M = card(A)$ and $c \in A$

The model was implemented in PyTorch, and training was carried out using the Google Colab platform. For ours Dataset of words Arabic of Al-Soudani script handwritten a minibatch size of 32 combined with a gradient accumulation factor of 2 was used,this makes the training process easier because potential "GPU OOM" (Out of Memory) errors on the graphics card are quickly detected at the beginning of the execution, ADAM [31] optimizer was employed with a fixed learning rate (α) of 0.0002, $\beta 1 = 0.9$ and $\beta 2 = 0.999$. was employed with a fixed learning rate.

However, after converting the training images to grayscale, caled down to 200-150 dots per inch, all images in each batch must have the same size. Therefore, we also set the size of all batches to have the same image size by padding smaller images with black pixels (0). Also, padding of the targets (labels) for each batch with a PAD token is performed to align them all to the same maximum length. The end-of-sequence symbol (EOS token) is also added to each padded sequence to indicate the end of the sequence. This allows the model to recognize the actual length of each sequence during training and make correct predictions. The size of alphabet $|A|$ is 38, shown in Figure. 9.



ء آ أ ؤ إ ئ ا ب ة ت ث ج ح خ د ذ ر ز س ش
ص ض ط ظ ع غ ـ ف ق ك ل م ن ه و ى ي

Fig 8: Unique characters in the dataset.

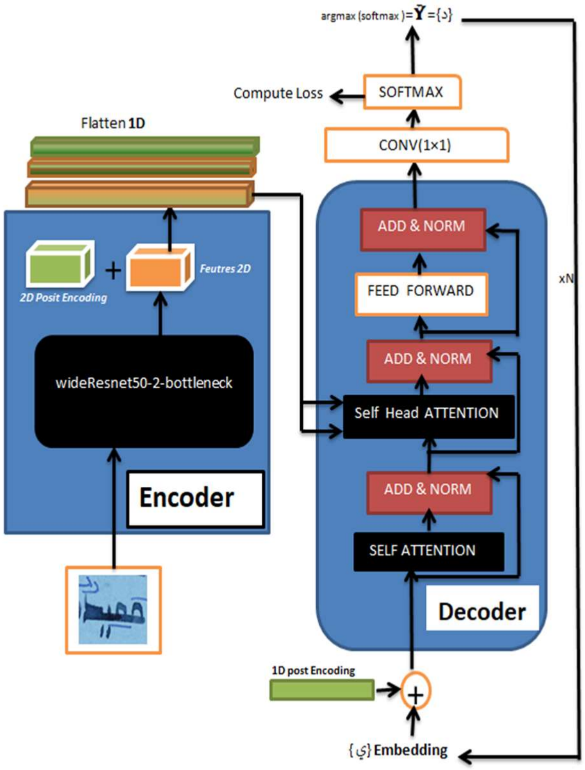


Figure9: Our architecture is made up of an wideResne-50-2 encoder, for the extraction of 2D features , and a transformer-based decoder for the recurrent prediction of the character/word د At each iteration , the model computes the representation of the current character/word token to recognize د based on the flattened features 1D and on the previous predictions.

5. EXPERIMENTS AND RESULTS

We trained our model on each of the three portions of the Al-Soudani script handwritten words dataset. Table II summarizes the results obtained. First, the model was trained on the dataset in a limited version of 475 images and after 158 training epochs; the model stopped and gave a CER of 58% and a WER of 88%. These very low results are due to the limited size of the training data.

Secondly, the model was trained on another version of, larger than the previous one, containing 3196 images. This time the model was stopped after 64 training epochs, he model gave a CER of 36% and a WER of 71% on the validation set. An improvement of metrics clearly registered.

Finally, we trained the model on the entire dataset which currently contains more than 30430 images, the model shows a high accuracy: only after 9 training epochs on the validation set, where it gave a 10 % CER and a 9.9 % WER. This is promising for the automatic recognition of lines, paragraphs and consequently pages of this Al-Soudani script which presents many challenges. (see Table I).

We remark that the CER is superior to WER and this is because of the omnipresent problem of character overlapping in the Al Maghribi Al-Suodani script.

Table Ii: Fine-Tuning With Different Portions Of Ours Dataset

Models	Dataset	#Epoch	CER	WER
Ours	(475)	158	58%	88%
	(3196)	64	36%	71%
	(30430)	9	10%	9.9%

Table IIi: Ablation Study On Convolutional Architectures

Encoder	#Epoch	CER	WER
Resnet18	64	87%	71.1%
Resnet34	64	100.4%	73.7%
Resnet50	64	100.1%	75.7%
WideResnet50_2	64	88%	70.4%

A. Ablation study

For the ablation studies, all experiments are trained from scratch using the dataset (3196), which consists of 3196 annotated word images and excludes synthetic data. Training is conducted for a fixed number of epochs for each experiment. Character Error Rates (CER) and Word Error Rates (WER) are calculated on the validation set. These CER and WER values are also employed as indicators for hyperparameter selection, as demonstrated in Table V, Table IV.

1 - The CNN Feature Encoder: We explored different popular Convolutional Neural Network architectures (ResNet and WideResNet) for the feature encoder . The best results were obtained

with wideResNet models. We trained the model using various encoder configurations. According to Table III, the best performances are achieved with a version of wideResNet50.

2 - The Transformer Decoder: Initializing the initial weights for deep learning models is crucial to achieve good performance. For this purpose, we experimented with various weight initialization methods to initialize the initial weights of the Decoder, while keeping the initial weights of the Encoder fixed throughout all experiments. Table IV shows the results of this study.

We found that the Kaiming [32] weight initialization method yields better performance than the other methods. The Xavier [33] method ranks last, after the methods using default initial neuron weights, which are often employed by state-of-the-art models in open-source deep learning platforms (in this ablation study, we utilize PyTorch). Interestingly, the normal random initialization method seems to come in second place in terms of WER rates. Despite the study being conducted for a limited number of epochs, the obtained results capture the general trends of these weight initialization methods, even when training the modules for longer durations.

B. Comparison with FPHR Model

To ensure a fair comparison between the proposed architecture and the Transformer-based FPHP [26] model, we reimplemented the FPHP model and subsequently trained and evaluated both the FPHP model and our model on the dataset, while keeping the exact same conditions. First and foremost, in Table V, we present the Character Error Rate (CER) and Word Error Rate (WER) on the our dataset test set. We also provide the model size and the time required during training. Despite the fact that the FPHR model has fewer parameters than ours, using only 50 layers in the encoder for both models, it also manages to achieve a slight reduction in training time compared to ours. However, our model achieves better performance in terms of CER and WER than FPHR, as shown in Table V, both models follow the same trend. The more real training data is available, the better the performance. Overall, the method based on our WideResNet50 encoder performs better than the FPHR method, except in the extreme case where only 475 annotated real training images are available, leading FPHR to achieve a better Character Error Rate (CER) than ours. The approach based on WideResNet50, being a much larger model, struggles in such conditions of drastic data scarcity. However, it should be noted that if we extend the FPHR encoder to 152 layers (ResNet-152), the model based on WideResNet50 as an encoder will take less time than FPHR during training and will likely achieve better performance.

Table Iv: Ablation Study On Initial Weights

Weight Initialization Methods	#Epoch	CER	WER
Xavier Initialization	10	100.42%	96.9%
Kaiming Initialization	10	100.41%	**95.5%**
Random Initialization	16	100.53%	96.2%
Default initial weights	19	100.37 %	96.8%

Table V: Comparison Between Fphr And Ours Model

Models	Dataset	#Epoch	CER	WER
Ours	(475)	158	58%	**88%**
Ours	(3196)	64	**36%**	**71%**
Ours	(30430)	9	**10%**	**9.9%**
FPHR(Encoder Resnet50) [26]	(475)	158	56%	90%
FPHR(Encoder Resnet50) [26]	(3196)	64	42%	76%
FPHR(Encoder Resnet50) [26]	(30430)	9	13.4%	12.8%

6. CONCLUSIONS

In this article, we proposed a new Encoder-Decoder model, where the decoder is a WideResnet50_2, this allows the model to capture more input features, surpassing other encoders based on Resnet, and a Transformer decoder for recognizing handwritten words in the Al-Soudani script. We conducted a comprehensive analysis and evaluation of this model, demonstrating the relevance of the proposed approach. Indeed, the presented results provide evidence that our method achieves state-of-the-art performance. We have proposed also a first version of a new dataset comprising 30,430 annotated images of handwritten words in the Al-Soudani Arabic Maghrebi script, with the aim of stimulating the scientific community to conduct research into the analysis and recognition of this challenging script. Our dataset has the particularity of taking into account the overlap between words, which is a major source of average recognition rates in ancient Arabic manuscripts. Our future work will first focus on augmenting and extending the dataset to cover the recognition of lines and different text areas in a page. Next, we will train our models on our extended dataset and improve them to achieve acceptable recognition rates that can aid in the automatic transcription of manuscripts written in Al-Soudani.

We intend as well, in our future work, to further enrich the dataset in order to balance the number of character occurrence.

REFERENCES:

[1] Alghamdi, Mansoor, and William Teahan. 2017. “Experimental Evaluation of Arabic OCR Systems.” PSU Research Review 1(3): 229–241. United Kingdom: Emerald Publishing Limited. DOI: https://doi.org/10.1108/PRR-05-2017-0026.

[2] Khan, F., Bouridane, A., Khelifi, F., Almotaeryi, R., and Almaadeed, S.(2014). Efficient segmentation of sub-words within handwritten Arabic words. International Conference on Control, Decision and Information Technologies

[3] Grüning, T., Labahn, R., Diem, M., Kleber, F., Fiel, S.: READBAD: A new dataset and evaluation scheme for baseline detection in archival documents. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 351–356. IEEE (2018). https://doi.org/10.1109/das.2018.38.

[4] Murdock, M., Reid, S., Hamilton, B., Reese, J.: ICDAR 2015 competition on text line detection in historical documents. In: 2015 13th International Confer ence on Document Analysis and Recognition ICDAR).pp.1171–1175.IEEE (2015).https://doi.org/10.1109/icdar.2015.7333945.

[5] Lawgali, A. (2015). A survey of arabic character recognition. International Journal of Signal Processing, Image Processing and Pattern Recognition., 8(2), 401–426. https://doi.org/10.14257/ ijsip.2015.8.2.37.

[6] Kassis, M., Abdalhaleem, A., Droby, A., Alaasam, R., and El-Sana, J. (2017a). Vml-hd: The historical arabic documents dataset for recognition systems. In 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), pages 11–14.

[7] Pantke, W., Dennhardt, M., Fecker, D., Mrgner, V., and Fingscheidt, T.: An historical hand written arabic dataset for segmentation-free word spotting - hadara80p. In 2014 14th International Conference on Frontiers in Handwriting Recognition, pages 15–20.

[8] Kiessling, B., Ezra, D.S.B., Miller, M.T.: BADAM:A Public Datasetfor Baseline Detection in Arabic-Script Manuscripts. In: Proceedings of the 5th International Workshop on Historical Document Imaging and Processing. p. 13–18. HIP ’19, Association for Computing Machinery (2019).

[9] Chahan Vidal-Gorène et. al., “ RASAM – A Dataset for the Recognition and Analysis of Scripts in Arabic Maghrebi ” dans : Elise H. Barney-Smith, Umapada Pal (eds.), Documents Analysis and Recognition– ICDAR 2021 Workshops, Lecture Notes in Computer Science 12916, Springer, 2021, p. 265-281. 2021

[10] Moghaddam, R. F., Cheriet, M., Adankon, M. M., Filonenko, K., and Wisnovsky, R. (2010). IBN SINA: a database for research on processing and understanding of Arabic manuscripts images. In DAS’10: Proceedings of the 8th IAPR International Workshop on Document Analysis Systems, pages 11–18, New York, NY, USA. ACM.

[11] Shahkolaei, A., Beghdadi, A., Al-maadeed, S., and Cheriet, M. (2018a). Mhdid: A multi-distortion historical document image database. In 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pages 156–160.

[12] Shahkolaei, A., Nafchi, H. Z., Al-Maadeed, S., and Cheriet, M. (2018b). Subjective and objective quality assessment of degraded document images. of Cultural Heritage, 30:199–209.

[13] Adam, K., Baig, A., Al-Maadeed, S., Bouridane, A., El-Menshawy, S.: KERTAS: dataset for automatic dating of ancient Arabic manuscripts. International Journal on Document Analysis and Recognition (IJDAR) 21(4), 283–290 (2018).

[14] Lamia Berriche & Abeer Al-Mutairy (2020) Seam carving-based Arabic handwritten sub word segmentation, Cogent Engineering, 7:1, 1769315,DOI:10.1080/23311916.2020.1769315

[15] U. Marti and H. Bunke. The IAM-database: An English Sentence Database for Off-line Handwriting Recognition. Int. Journal on Document Analysis and Recognition, Volume 5, pages 39 - 46, 2002.

[16] Kiessling, Benjamin, Gennady Kurin, Matthew Miller, and Kader Smail. 2021. "Advances and Limitations in Open Source Arabic-Script OCR: A Case Study." Digital Studies/Le champ numérique 11(1): 8, pp. 1–30. DOI: https://doi.org/10.16995/dscn.8094.

[17] G. Hinton, Neural Networks for Machine Learning -Lecture 6a - Overview of mini-batch gradient descent., 2012, doi:10.1017/9781139051699.031.

[18] Mohamed Ibn Khedher, Houda Jmila, Mounim El Yacoubi. Automatic processing of Historical Arabic Documents: a comprehensive survey. Pattern Recognition, Elsevier, 2020, 100, pp.1071441:10714417.ff10.1016/j.patcog.2019. 107144ff.ffhal 02481354.

[19] Alexandre, P. ,Langue et langage en Afrique noire,Paris,Payot, 169p. (Introduction) (1967).

[20] Mamadou Cissé, Ecrit en écriture en Afrique de l'Ouest , Revue électronique internationale des sciences du langage Sud Langues No 6 juin 2006.

[21] Ngom Fallou, Castro Eleni, Diakite Ablaye. (2018) African Ajami Library: EAP 1042 Digital Preservation of Mandinka Ajami Materials of Casamance, Senegal. Boston: Boston University Libraries:http://hdl.handle.net/2144/27112 https://hdI.handle.net/2144/37430 Boston University.

[22] Octave Victor Houdas, Essai sur l'écriture maghrebin Ecole des langues orientales vivantes, 1886 31 pages.

[23] Ba, Jimmy Lei and Kiros, Jamie Ryan and Hinton, Geoffrey E.Layer Normalization, arXiv 2016,https://doi.org/10.48550/arxiv.1607.06450 , https://arxiv.org/abs/1607.06450.

[24] Chorowski, Jan and Bahdanau, Dzmitry and Cho, Kyunghyun and Bengio, Yoshua End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results, arXiv2014,https://doi.org/10.48550/arxiv.1412. 1602.

[25] Vaswani A. and Shazeer N. and Parmar N.and Uszkoreit J. and Jones L. and Gomez A.N. and Kaiser L. and Polosukhin, I. Attention is all you need. CoRR, abs/1706.03762, 2017.

[26] S. S. Singh and S. Karayev, "Full page handwriting recognition via image to sequence extraction," in International Conference on Document Analysis and Recognition, 2021. 4, 5.

[27] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. CoRR, abs1512.03385, 2015.

[28] Zagoruyko, Sergey and Komodakis, Nikos, Wide Residual Networks ,arXiv 2016,arXiv.org https://arxiv.org/abs/1605.07146.

[29] Parmar, N., Vaswani, A., Uszkoreit, J., Lukasz Kaiser, Shazeer, N., Ku, A., and Tran, D. Image transformer, 2018. ArXiv.

[30] Hendrycks, D. and Gimpel, K. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. CoRR, abs/1606.08415, 2016. URL http://arxiv.org/abs/1606.08415.

[31] Kingma, D.P. and Ba, J. Adam: A method for stochastic optimization, 2017.

[32] Kaiming He, and Xiangyu Zhang, and Shaoqing Ren, and Jian Sun Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, 2015 arXiv.

[33] Xavier Glorot, Yoshua Bengio Understanding the difficulty of training deep feedforward neural networks ,Journal of Machine Learning Research January 2010.

[34] Alaasam, R., Barakat, B. K., and El-Sana, J. (2018). Synthesizing versus augmentation for arabic word recognition with convolutional neural networks. In 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pages 114–118.

[35] Alaasam, R., Kurar, B., Kassis, M., & El-Sana, J. (2017, April). Experiment study on utilizing convolutional neural networks to recognize historical Arabic handwritten text. In 2017 1st International Workshop on Arabic script analysis and recognition (ASAR) (pp. 124-128). IEEE.

[36] Hassen, Hanadi, Somaya Al-Madeed, and Ahmed Bouridane. "Subword Recognition in Historical Arabic Documents using C-GRUs." *TEM Journal* 10.4 (2021).