Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

AN ENHANCED WORD LEVEL ARABIC OCR BASED ON DUAL ENCODER TRANSFORMER ARCHITECTURE

Khulood Gaashan[1] and Maram Bani Younes[2]

(Received: 16-Jun.-2025, Revised: 12-Jul.-2025 and 12-Aug.-2025, Accepted: 13-Sep.-2025)

ABSTRACT

Arabic script is one of the most sophisticated and difficult scripts. It uses different shapes of characters with complex diacritical marks that are difficult to distinguish from the dots of characters. This script's distinctive features make the Optical Character Recognition (OCR) procedure more challenging and result in low-accuracy recognition. Different studies have aimed to introduce high-accuracy Arabic OCR in the literature. However, enhancing the accuracy of reading the words has been an open issue that depends on the used dataset and the developed recognition model. Besides, considering diacritics has been limited and not sufficiently addressed. Experimental tests on words with diacritics in prior models have shown bad accuracy that does not exceed 60%. Consequently, this work aims to introduce a new, accurate deep-learning model for Arabic OCR that considers words with and without diacritical marks. It utilizes a dual encoder transformer (DTrOCR), a deep-learning architecture that has demonstrated superior performance in identification and classification tasks. The proposed DTrOCR creates multi-batch sizes. It has been trained using a comprehensive, generated Arabic word-based dataset named MFSRHRD and tested on unseen datasets. The accuracy of configuring Arabic words without diacritics reaches 98.5%. However, for words with diacritics, it achieved an accuracy of 89.9%.

KEYWORDS

Arabic OCR, Multi-batch size, Transformer, Dual encoder transformer, Decoder, Feature extraction, Self-attention mechanism.

1. INTRODUCTION

Optical character recognition (OCR) is a sub-discipline of pattern recognition and computer vision. OCR has received more and more attention and has become a popular and promising research area in computer and pattern-recognition communities. However, recognizing documents with Arabic text contents is a popular and actively developed field. The main objective of the OCR system is to convert the images of a document, whether printed or hand-written, into computer-editable text to generate digital copies of text documents [22], [7], [17]. Moreover, OCRs have several applications, such as archive organization, automated plate recognition and automated ticketing [10].

The process of Arabic OCR encounters several challenges due to the distinctive features of the Arabic script. Arabic is a right-to-left written language that consists of twenty-eight letters. Each letter can have several forms based on its place inside a word, whether at the beginning, middle, end, or isolated. The language does not differentiate between capital and lowercase letters. Dots, also known as "Ijam," and diacritical markings, also known as "Tashkeel," introduce complication by distinguishing letters and modifying the meanings of words. In addition, the Hamza can occur in several locations and it can be challenging to differentiate letters with similar structures, such as Saad and Taa. Characters such as Raa, Dal and Waw, which do not form a connection with the letter that follows them, add to the complexity of separating words, …etc.

Researchers have recently used artificial-intelligence (AI) mechanisms to recognize Arabic characters, words and printed texts. The deep-learning models have become a significant player in Arabic language recognition. Previous studies have encountered several common challenges, such as a lack of diverse and balanced datasets, rare diacritics considerations, limited model accuracy, especially when dealing with diacritics and the model's capacity to generalize [25], [22], [9], [7].

To address these specific challenges, our motivation behind using a Dual Encoder Transformer is to allow the model to learn from both global and local patterns in the input images. In Arabic, very small marks, such as diacritics or dots, can completely change the meaning of a word. Traditional models with a single encoder may struggle to capture both overall word structure and fine-grained details at

1. K. Gaashan is with Software Engineering Department, Philadelphia University, Amman, Jordan. Email: khuloodgaashan@gmail.com
2. M. Bani Younes is with Faculty of Information Technology, American University of Madaba, Jordan. Email: m.baniyounes@aum.edu.jo

the same time. By using two encoders with different patch sizes, our model can analyze the broad shape of the word and also focus on subtle features like diacritics. This dual-path design makes it more capable of handling the complex nature of Arabic script and improves recognition accuracy, especially in the presence of diacritics.

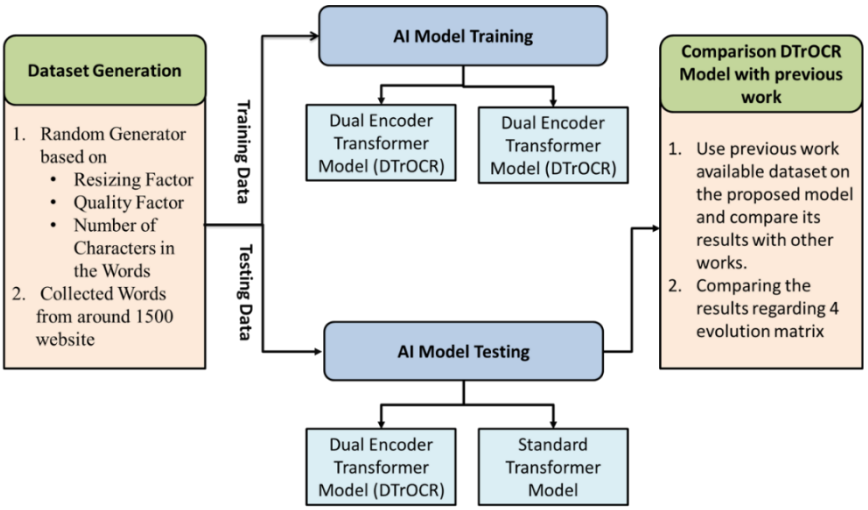

Figure 1. Test and train DTrOCR model.

We propose a Dual Encoder Transformer (DTrOCR) model in this work. The key feature of the proposed model is its ability to process input images by splitting them into multi-batch sizes instead of single-batch sizes, enabling more effective feature extraction. It was trained on our previously proposed comprehensive generated Arabic dataset MFSRHRD [6]. This dataset, created specifically for Arabic characters and word recognition, was chosen for its high number of records, comprehensive coverage and diversity of font sources. This dataset also includes many Arabic words with diacritics, which most previous Arabic datasets have neglected. The dataset was divided into 80% for training and 20% for testing, ensuring controlled conditions for model evaluation.

Figure 1 illustrates the general flowchart for training and testing the proposed model. As we can see from the Figure, besides testing the performance of the Dual Encoder Transformer, a standard Transformer model was trained on the same dataset [6]. This is to establish a performance baseline and illustrate the benefits of the proposed model. The Dual Encoder Transformer outperformed the standard model, achieving higher accuracy and generalization across different fonts and writing styles. The model's performance was evaluated using four main metrics: accuracy, precision, recall and F1-score. Moreover, the proposed model (DTrOCR) was further tested on other unseen datasets, including APTI, IFN/ENIT and MMAC, where it continued to demonstrate high performance compared to previous OCR models, showcasing its robustness and effectiveness.

In general, the main contributions of this work are summarized by:

- Introducing an enhanced Arabic OCR system using a Dual-encoder transformer (DTrOCR), which improves the features extracted from images using multi-batch sizes instead of a single-batch size.
- Testing the proposed model's efficiency and generalization in terms of recognizing unseen Arabic words with and without diacritics.

The rest of this paper is structured as follows: Section 2 investigates previous deep-learning techniques for Arabic OCR. It illustrates the main considerations and specifications of these models. Then, it identifies the main gaps and the needed work in this field of research. Section 3 introduces the details of the proposed Arabic OCR model (DTrOCR). It outlines the main phases, considerations and steps of the model. Section 4 presents the details of the training and testing processes, including the experimental setup and sequential phases. Section 5 evaluates the performance of the Arabic OCR model on unseen datasets and tests its efficiency. Section 6 concludes the entire paper with a summary of key findings, directions for future research and recommendations for researchers in this field of study.

2. RELATED WORK

Optical Character Recognition (OCR) technology has been widely used over the past few decades, aiming to transform images of text into editable texts. It mainly aims to digitize and process text information. Regarding Arabic OCR, the challenges are more complicated due to the complex nature of the Arabic script, as discussed earlier. This section provides an overview of the previous studies that developed Arabic OCR. It identifies the main methodologies and challenges in this field of research, tracing the main progress made and highlighting the gaps and required work.

Previous Arabic OCRs have been developed using several deep-learning and artificial-intelligence algorithms. First, several studies [7], [19], [15] have combined multiple popular and strong deep-learning techniques, such as convolutional neural network (CNN), Long Short Term Memory (LSTM) and connectionist temporal classification (CTC), …etc. Other studies [16], [14], [13] have considered the Generative Adversarial Networks (GANs) to enhance the quality of the OCR models. Recently, researchers started using transformers to develop more accurate Arabic OCR [23], [7].

Several researchers have combined multiple machine/ deep-learning algorithms to obtain accurate Arabic OCRs. For instance, Z. Noubigh et al. [21] presented a model that combines the CNN, Bidirectional Long Short-Term Memory (BLSTM) and CTC algorithms. The model was trained and tested using the KHATT and HACDB datasets [26] to classify Arabic characters. It achieved high acceptable accuracy and the character error rates (CERs) of this model were 2.74% and 2.03% on the KHATT and HTID datasets, respectively.

Dahbali, Aboutabit and Lamghari [4] proposed a hybrid model combining CNN with attention (CBAM) and BLSTM to improve Arabic handwritten script recognition. The model integrates spatial and sequential features using Connectionist Temporal Classification (CTC) decoding and data augmentation. It was evaluated on the KHATT dataset and achieved significant improvements in recognition accuracy, outperforming prior approaches. Al-Taani and Ahmad [18] used Residual Neural Networks (ResNet) for Arabic handwritten character recognition. Their model was tested on several benchmark datasets including MADBase, AIA9K and AHCD. The approach achieved up to 99% accuracy, demonstrating the benefit of deep residual learning architectures in handling the variability of Arabic handwriting.

Shtaiwi et al. [11] proposed an end-to-end machine-learning solution for recognizing handwritten Arabic documents that combines several deep-learning models, resulting in improved robustness and accuracy on real-world datasets. [11] proposed an Arabic OCR model that combined Convolutional Recurrent Neural Network (CRNN) and BLSTM. This model aims to recognize Arabic handwritten content based on the character unit as well. It was trained on the MADCAT dataset [24] and achieved a CER of 3.96%. On the other hand, M. Boualam et al. [15] proposed an OCR model that combined CNN, RNN and CTC models. The proposed OCR model aims to recognize text from handwritten village names in Tunisia using the IFN/ENIT dataset [30]. This model classifies Arabic characters and Arabic words. It achieved a CER of 2.10% and a word error rate (WER) of 8.21%. Fasha et al. [22] merged CNN and BLSTM with a CTC loss function, achieving a character recognition rate (CRR) of 98.76% and a word recognition rate (WRR) of 90.22%.

Generative adversarial networks (GANs) have also been explored to improve the performance of Arabic OCR. GANs do not directly perform character recognition in OCR; they can significantly enhance the performance and robustness of OCR systems. This is achieved by improving data quality and increasing the diversity of training datasets. Several studies have been introduced using this mechanism in the literature. First, Y. Alwaqfi et al. [16] early explored GAN-based models. M. Eltay et al. [14] also utilized GANs for adaptive augmentation. This model achieved an accuracy of 95.51% on the character-recognition level and 89.52% on the word-recognition level. Moreover, A. Mostafa et al. [17] faced challenges in ensuring the quality of generated samples, particularly in dealing with connected Arabic letters, making the results less conclusive with an accuracy that reached 95.08%. S. Jemni et al. [13] demonstrated high proficiency in Arabic and English OCR using GANs with a hybrid model CNN-RNN-CTC, which achieved 75.6% accuracy. However, complexities in combining deep-learning models at each stage were reported in this model.

Recently, transformer-based models have become increasingly popular for Arabic-text recognition. A. Mustafa et al. [17] developed a specialized dataset using thirteen unique web typefaces, including

Table 1. Previous Arabic OCR deep-learning models with their limitations.

Ref.	Technique	Dataset	Recognition Level	ACC	Description	Limitation(s)
[21]	CNN-BLSTM-CTC	KHATT, AHTID	Word-level	97%	Combined CNN for feature extraction and BLSTM for sequence prediction	The test was conducted on 901 samples from AHTID dataset
[11]	CRNN-BiLSTM	MADCAT	Paragraph-level	96.04%	CRNN-BiLSTM model integrates Convolutional Recurrent Neural Networks (CRNN) with Bidirectional Long Short-term Memory (BiLSTM) layers	Requires significant time to train due to additional layers of BiLSTM
[15]	CNN-RNN-CTC	IFN/ENIT	Word-level	97.90%	CNN layers for feature extraction, followed by RNN layers for sequential data processing	No real testing was performed of generalization
[22]	CNN-BiLSTM-CTC	Custom dataset	Word-level	98.76%	Five CNN layers for feature extraction and two BiLSTM layers for sequence prediction	When tested on noisy images, the accuracy drops drastically to 22.71 CER and 85.82% WER
[16]	GANs-CNN	AHCD	Character-level	99.78%	GANs for data augmentation with CNN for classification	Error rate, training and testing ratios and number of images after applying augmentation were unclear
[14]	GANs-BiLSTM	IFN/ENIT, AHDB	Word-level	95.8%	ScrabbleGAN for augmentation followed by BiLSTM for recognition	Still suffers when tested on challenging test sets
[17]	CDCGAN-CNN	AHCD	Character-level	95.08%	GANs for data generation combined with CNN for classification	Generation of characters with dots is still a challenging task. Additionally, classes can be unbalanced
[13]	GANs-CNN-RNN-CTC	KHATT	Word-level	75.6%	The generator employs U-net, while the discriminator uses CNN. For OCR model, CNN, followed by two layers of Bi-GRU and then a CTC layer, are stacked	Very complex model in each step of the OCR process, yet it produces an accuracy close to that of the baseline model
[17]	CNN-Transformer	Custom dataset, KHATT	Word-level	92.7%	ResNet101 combined with Transformer for sequence processing	Shortage of resources and computational power; complex model
[12]	Transformer With ross-attention	Custom dataset, KHATT	Word-level	81.55%	Transformer architecture with cross-attention mechanisms for better feature extraction	Some weights were randomly initialized, leading to a limited Accuracy
[8]	Transformer	KHATT	Word-level	97.7%	Transformer architecture for both image understanding and wordpiece-level text generation	Synthesized dataset for pre-training consists of 2.2 M images, which is relatively small compared to other methods that use hundreds of millions of images
[1]	QARI-OCR (vision-language model)	Diacritics-rich synthetic +real printed Arabic texts	Page/Word-level	98%	Vision-language Model fine-tuned from Qwen2-VL, optimized for Arabic script and diacritic handling	Limited font variety, fixed font size, no handwriting support; mostly trained on synthetic data
[3]	HATFormer (Transforme based HTR)	Historical r-handwritten Arabic manuscripts dataset	Line/Word-level	95%	Transformer encoder-decoder customized for historical Arabic handwriting with diacritic support	Small training dataset; still moderate error rate; limited to historical handwritten text
[2]	Hybrid CNN+ Transformer OCR system	Printed & Handwritten Arabic text (with digits)	Character & Word	99.4%	A hybrid CNN-Transformer OCR system with excellent accuracy for printed Arabic text (CER = 0.59%) and competitive performance on handwriting (CER = 7.91%). It also includes effective text detection (F-measure 79%).	Still struggles with handwritten Arabic (higher WER) and shows sub-optimal text detection performance on complex backgrounds or irregular handwriting.

hand-written samples from the KHATT dataset. The proposed model employed a two-part system: a CNN for feature extraction and a transformer model with four encoders and decoders. This model considers character-level and word-level configurations and achieves a CER of 7.27% and a WER of 8.29%. Besides, Momeni et al. [12] examined two types of transformers: transformer transducer and transformer with cross-attention, using a synthesized dataset of 500,000 printed Arabic images and the KHATT dataset for testing. The transformer with cross-attention achieved a CER of 18.45%, outperforming the transformer transducer, which achieved 19.76% CER. Most studies using transformers for Arabic OCR, such as ALNASIKH [8] and OCFormer [17], have leveraged standard transformer models, like TrOCR or vision transformers. These existing models primarily focus on utilizing single encoder-transformer architectures for Arabic handwritten and printed text recognition.

Table 1 systematically illustrates the main considerations and limitations in the previous studies in this field of research. After reviewing earlier studies in this field of research, it is clear that transformer-based models are emerging as strong contenders. This is due to their attention mechanisms and ability to process long sequences. A dual encoder transformer has been used in the medical field [5]. However, it has not been applied before for the Arabic OCR. In recent years, researchers have made many improvements to the standard transformer model to enhance its performance in recognizing Arabic characters. These changes have focused on the encoder and decoder parts of the model. However, one challenge remains the issue of a single batch size during training, which can limit efficiency. A promising solution to this problem is using multi-batch size inputs. To achieve this, we employ a dual encoder transformer, a method that has been successful in the medical field for image recognition. In this research, we apply this approach to improve the recognition of Arabic characters. Moreover, we aim to investigate the ability of the Dual Encoder to be implemented to recognize Arabic letters and words.

3. THE ENHANCED PROPOSED ARABIC OCR (DTROCR)

In this work, we developed a deep-learning model for recognizing Arabic letters and words using a dual encoder transformer. As shown in Figure 4, the model consists of two different encoders, each designed to extract unique features from the input data based on different batch sizes. The outputs from these encoders are combined using a fusion mechanism based on two multi-head attention layers. This step helps the model merge the features effectively, ensuring that the most important information is retained. Then, the fused features are sent to the decoder, which produces the final output. This approach helps the model better understand the complex nature of Arabic script. It highlights the role of the external fusion layer in achieving more accurate recognition results. The input image is divided into multiple-batch sizes (N) using two dual encoders. The model partitions the input image based mainly on its size, as illustrated in Equation (1).

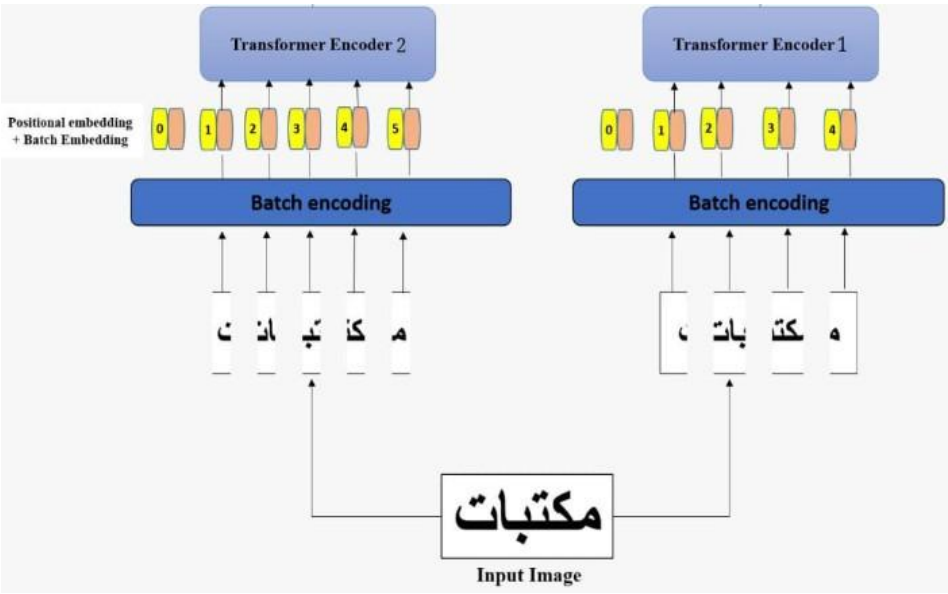

Figure 2. Multi-batch sizes in DTrOCR model.

$$N = \frac{HW}{P^2} \quad (1)$$

where: *N:* is the total number of batches; *H:* is the height of the input image, *W:* is the width of the input image and *P:* is the batch size.

Each encoder has multiple attention layers, enabling the extraction of diverse features from the same image. Using dual encoders with varying multi-attention layers allows the model to effectively extract a wide range of features from the same input image. This enhances the overall performance in Arabic optical character recognition.

Figure 2 illustrates the process beginning with an input image of an Arabic word (مكتبات). The image is divided into batches of multiple sizes, with each patch fed into one of the two different encoders in our model. Each encoder processes the batches independently, capturing unique features. The outputs from both encoders are then combined using a multi-head attention mechanism, allowing the model to learn more nuanced details from each perspective. This multi-batch-size approach enhances OCR accuracy by leveraging two unique views of the data, making it especially effective for recognizing the complex characteristics of Arabic script.

To better illustrate how the proposed dual encoder architecture processes input images with different batch sizes, as illustrated in the proposed model (Figure 4), the dual-encoder architecture extracts two complementary types of features from the input image. Encoder 1 (E1), which operates on smaller patch sizes, focuses on capturing fine-grained local features, such as character edges, diacritical marks and subtle variations between visually similar characters. This detailed representation allows the model to disambiguate closely related characters with high precision. In contrast, Encoder 2 (E2), which processes larger batch sizes, extracts global and coarse-grained features, capturing the overall word shape, inter- character spacing and distribution patterns. These higher-level features provide a more holistic view of the input, allowing the model to understand the context and structural layout of the word as a whole.

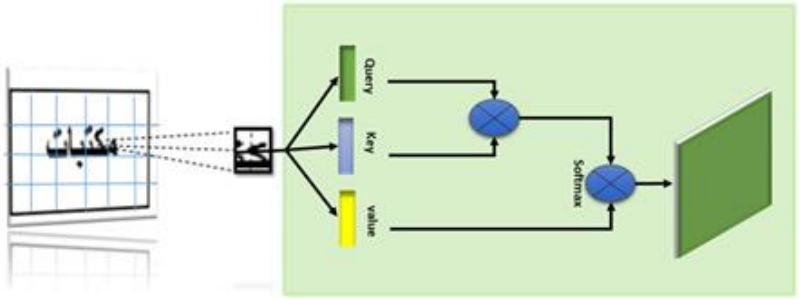

Figure 3. Self-attention mechanism.

To leverage the complementary strengths of both encoders, a feature-fusion mechanism based on multi- head attention is applied. This mechanism combines fine and coarse representations into a single rich feature representation, which is then passed to the decoder. The fused representation enables the decoder to reconstruct the textual output with improved accuracy, as it benefits from both local detail and global context simultaneously. This fusion strategy is a key component of the proposed model's superior performance compared to a standard single-encoder Transformer.

As shown, the input image is split into two versions of batch sequences—each with a different batch size. The first sequence (Input 1) uses larger batches and is passed to E1, which is optimized to capture high-level structural and global features of the word. The second sequence (Input 2) uses smaller batches and is sent to E2, which focuses on detecting fine-grained local features, such as diacritics and subtle visual differences between characters.

Both encoders (E1 and E2) process their respective inputs through multiple layers of multi-head attention and feed-forward networks. Their outputs are then merged using a two-stage multi-head attention fusion block. This external fusion mechanism allows the model to integrate complementary information from both encoders, resulting in a richer and more balanced feature representation. The fused features are then passed to the decoder for final output generation.

This multi-batch dual encoder design allows the model to effectively balance between general layout understanding and fine detail detection, which is particularly important for complex Arabic text with diacritics. Figure 4 illustrates the overall architecture of the proposed DTrOCR model and how the dual encoders collaborate using multi-batch sizes. The input image is divided into two different patch sets with varying sizes. These are then independently fed into two transformer-based encoders: E1 and E2. E1 receives larger batches and focuses on capturing global structural patterns of the word, while E2 processes smaller batches to detect fine-grained local details, such as dots and diacritics, which are essential for distinguishing similar Arabic characters. Each encoder applies multiple layers of multi-head attention and feedforward networks to extract high-level features from its respective input. The outputs from both encoders are then passed to a fusion module composed of stacked multi-head attention layers. This fusion stage is responsible for integrating both global and local information into a single comprehensive feature representation. The fused features are then sent to the decoder, which generates the final output text. This dual-encoder, multi-batch approach allows the model to learn rich and diverse features from the same input image, which significantly improves recognition accuracy.

The proposed model is discussed in the following steps:

1) ***Pre-processing Input Image:*** The model starts by preparing the input images of the Arabic words into a standard format that facilitates their usage in the proposed OCR model. All images are resized to a standard size of 512 x 512 pixels. Thus, the images are all of the same size, which makes them easier to process. Then, the pixel values are adjusted to a consistent range, which helps the model learn more effectively. Finally, the images are grouped into batches, which enables the model to handle several images at once. This makes the processing faster and more efficient.

2) ***Setting the Encoder Parameters:*** Using the dual-encoder model aims to improve the recognition of Arabic diacritics like ("damma," "kasra", …etc.) and small text features. Traditional OCR models miss these details, as diacritics are easy to overlook. Table 2 illustrates different hyper-parameters for two different decoders, depending on the task of each encoder. The first encoder is set to receive a larger batch size to understand the broader context and global features of the image. The second encoder receives a smaller batch size to focus on finer details and local features. After encoding, the latent representations from both encoders are concatenated to form a comprehensive feature vector using the multi-head attention-based integration concatenation mechanism.

Table 2. Hyper-parameter comparison between first and second encoders.

Hyper-parameter	**Encoder 1**	**Encoder 2**
Patch Size (p)	32 pixels	16 pixels
Embedding Dimension	512	768
Number of Layers	6	8
Number of Attention Heads	8	12
Feedforward Dimension	2048	3072
Dropout Rate	0.2	0.1
Batch Size	32	16
Calculated Patches (N)	256	1024
Learning Rate	1e-4	1e-4
Optimizer	Adam	Adam
Positional Encoding	Sinusoidal	Sinusoidal

3) ***Multi-head Attention Outside the Encoder:*** Final outputs of encoders E1 and E2 are merged before sending them to the decoder. Merging these outputs allows the model to gather features learned by both encoders. This gives the decoder a more comprehensive view and enables it to achieve more accurate results. Additionally, having Multi-Head Attention outside the encoder lets the combined features be processed again as a preparatory step before reaching the decoder. This extra step can help refocus attention on specific details from each input, improving the model's accuracy by ensuring that the most relevant information is emphasized.

Unlike models like TrOCR and OCFormer that use only one encoder with one fixed batch size, our model uses two encoders—each with a different batch size. One focuses on the big picture of the Arabic word, while the other looks closely at the small details, like dots and diacritics. This is very useful in Arabic, where small marks can completely change the meaning of a word. By combining what both encoders see, our model builds a better understanding of the word and makes more accurate predictions. This approach helps our model perform better than other models, especially when dealing with complex Arabic writing.

4) ***Self-attention Mechanism:*** The attribute that distinguishes transformers most is the self-attention mechanism. Figure 3 graphically illustrates the self-attention mechanism on an input image. In this work, the purpose of self-attention is to calculate the connections between various components of the feature vector to capture interdependency, following these sequential steps:

- Query, Key, Value (Q, K, V) Vectors: The feature vector is transformed into separate Q, K and V vectors using projection.
- Attention scores are calculated by taking the dot product of the query and key vectors and then applying a softmax operation to obtain attention weights.
- Context vector is derived by multiplying the attention weights with the value vectors, so highlighting significant characteristics.

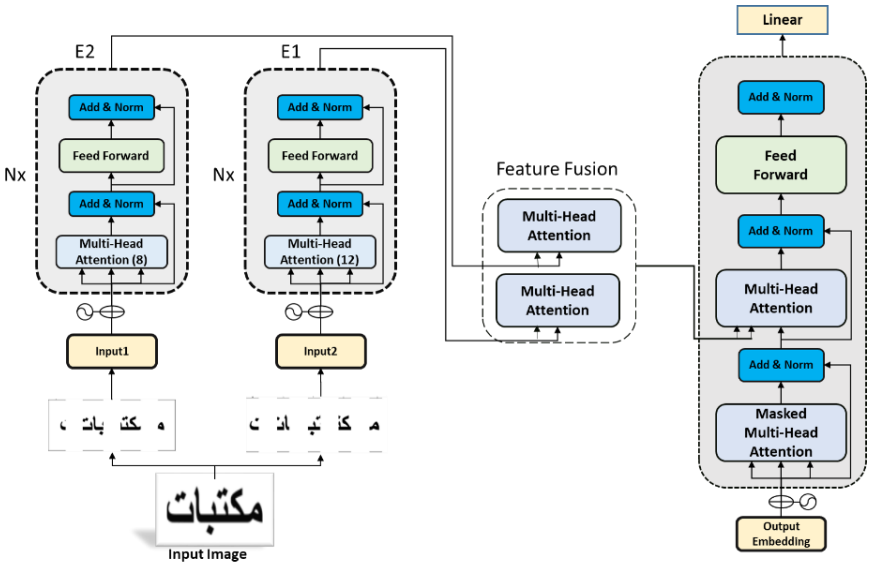

Figure 4. Proposed DTrOCR model.

4. Training and Testing the Proposed DTrOCR Model

The proposed DTrOCR model was trained using the Adam optimizer with a learning rate of 1e-4 and early stopping was applied to avoid overfitting. Hyper-parameter tuning was conducted to enhance the model's performance. The optimal configuration was determined through a grid-search approach, where various combinations of hyper-parameters were evaluated, selecting the configuration that achieved the highest validation accuracy for Arabic OCR tasks.

Careful tuning of hyper-parameters was essential to achieve the high performance of the proposed DTrOCR model. In particular, the choice of batch size, embedding dimensions and number of attention heads significantly influenced the model's ability to capture both global and local features of Arabic script. For Encoder 1, a larger batch size (32×32) and eight attention heads were selected to focus on extracting broader contextual information and global structural dependencies, which are critical for recognizing the general word shape and layout. Encoder 2 was configured with a smaller batch size (16×16) and twelve attention heads to focus on fine-grained details, such as diacritical marks and subtle character variations. This combination was determined through an extensive grid search, where different hyper-parameter configurations were compared based on validation accuracy and F1-score. Models with smaller batch sizes showed better generalization but slower convergence, whereas larger batch sizes improved training stability. The final configuration balanced these effects,

achieving faster convergence without sacrificing the ability to generalize to unseen datasets. As a result, the DTrOCR model achieved a 9.3% improvement in accuracy over the baseline Transformer, demonstrating the critical role of hyper-parameter selection in enhancing the model performance.

The dataset used for training the DTrOCR model was generated by our previous work [6]. It featured a diverse collection of Arabic text images that simulated various real-world scenarios, including different fonts, styles and a high number of recorders. Pre-processing techniques were applied to enhance the data's quality, such as noise reduction, using the same software used for the dataset generated to improve image clarity. These steps were essential to ensure that the model could accurately detect and recognize characters, even in challenging conditions, thus enhancing its overall robustness and generalization.

This generated dataset (MFSRHRD), which means Multiple Fonts, Sizes, Resources and High Records dataset, aims to fill the gaps in the available and open-source Arabic datasets. It faces all the challenges and weaknesses of the previously proposed datasets. Thus, the training process can be completed comprehensively. To obtain a model characterized by the generalization feature, the dataset must be large enough to include all possible scenarios in the trained machine. Figures 5, 6 show samples of the used dataset.

Figure 5. Samples of the used dataset.

Figure 6. Samples of characters with Tashkeel in MFSRHRD dataset.

In training the proposed DTrOCR model, the batch sizes for the two encoders were selected based on the objective of capturing both global and local features. Larger batches enable the model to extract structural information from the overall word shape, while smaller batches allow it to focus on fine-grained details, such as diacritics. Regarding the number of attention heads, a hyper-parameter tuning strategy using grid search was employed, where multiple configurations were evaluated and the one yielding the highest validation accuracy on the development set was chosen. This approach is commonly used in deep-learning research to ensure optimal model performance.

4.1 Technical Details

In this sub-section, we will provide more technical details regarding the training environment and implementation setup. The proposed model was trained using an NVIDIA RTX 3090 GPU with 24GB of VRAM. Although the dataset is large, we handled it efficiently by using mini-batch training and a custom-data generator that loads image batches on-the-fly from disk during training. The total training time was approximately 120 hours.

The model was trained for 100 epochs with early stopping to avoid overfitting. Input images were pre-processed through resizing (512×512 pixels), normalization and noise reduction to ensure data quality and consistency. A detailed configuration of the encoder layers, batch sizes and attention mechanisms is already summarized in Table 2.

Furthermore, we provide open access to our dataset-generation software and a sample of the generated MFSRHRD dataset to encourage reproducibility. These resources are available on GitHub at: https: //github.com/KhuloodGaashan/arabic-ocr-dataset.

Moreover, three other datasets for testing (IFN/ ENIT [30], APTI [28], MMAC [27]) were used to check our model generalization and performance using unseen datasets, printed and handwritten. Figure 7 illustrates samples of these datasets.

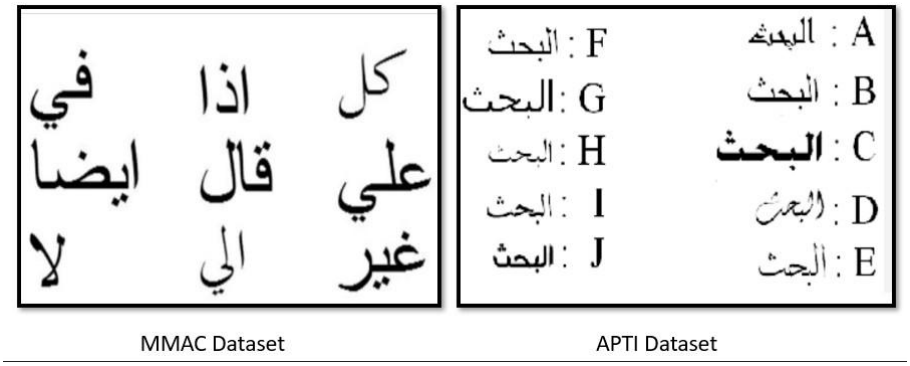

Figure 7. Samples from APTI & MMAC datasets.

5. PERFORMANCE EVALUATION

We evaluated the DTrOCR model using the MFSRHRD dataset and other unseen datasets. We employ several standard evaluation metrics to measure the performance of our Arabic OCR system. These metrics include accuracy, precision, recall, F1-score and the confusion matrix [20]. The details and equations used to compute these metrics are presented below:

- Accuracy: It measures the overall correctness of the system's predictions by calculating the ratio of correctly classified instances to the total instances.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

- Precision: It quantifies the proportion of correctly predicted instances out of all the instances predicted as Arabic OCR.

$$Precision = \frac{True\ Positives}{True\ Posiives + False\ Positives}$$

- Recall: It measures the proportion of correctly predicted Arabic OCR instances out of all the actual Arabic OCR instances.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

- F1-score: It provides a balanced measure of precision and recall, taking into account both metrics to evaluate the system's performance.

$$F1 - score = 2.\frac{Precision\ .Recall}{Precision + Recall}$$

5.1 Results & Analysis

Recognizing Arabic words using the DTrOCR model achieved outstanding results compared to previous models [22], [15], [8], [12]. The recognition accuracy reached 99.3%, demonstrating a significant improvement. The proposed model (DTrOCR) consistently outperformed previous methods in terms of recognition accuracy. This confirms its ability to recognize and thus accurately enhance Arabic OCR systems.

Figure 8 presents a comparison of the accuracy of various models, with our proposed model, DTrOCR (2024), achieving the highest accuracy at 99.30% compared to previous models. These results highlight the superior performance of DTrOCR (2024) in enhancing OCR accuracy.

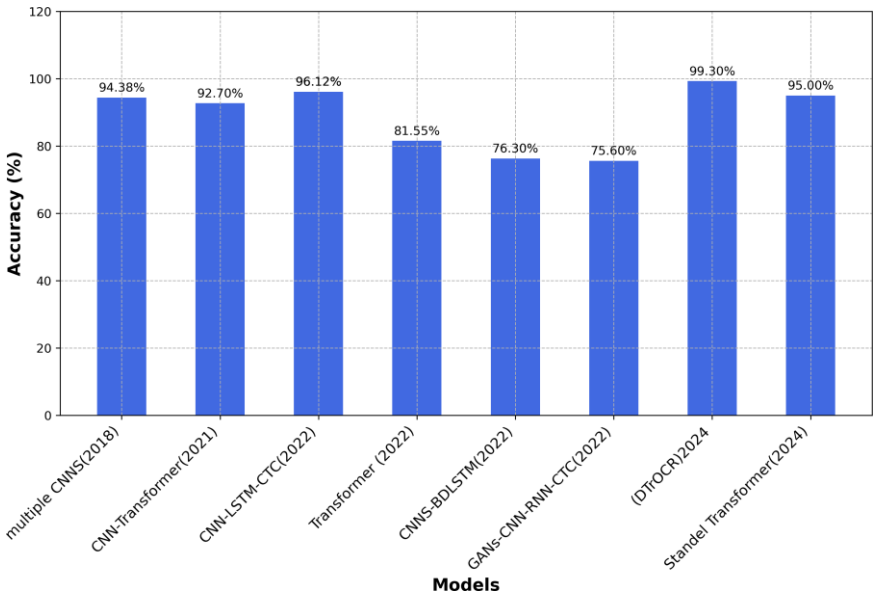

Figure 8. Comparing the accuracy of existing models with that of the DTrOCR model.

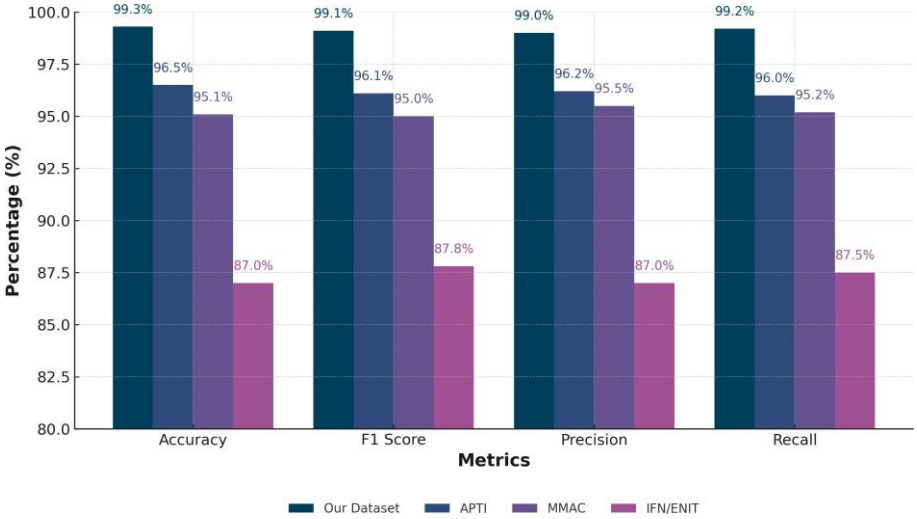

Figure 9. Evaluation matrix for the DTrOCR model using different datasets.

Using our previously generated dataset MFSRHRD, we trained two models, the first model was the DTrOCR Transformer and the second model was the Standard Transformer on the same dataset. The DTrOCR achieved higher and better results than the Standard Transformer and the results are illustrated in Table 3.

Table 3. Comparison between DTrOCR and standard transformer.

Model	Accuracy	F1-Score	Precision	Recall
DTrOCR	99.3%	99.1%	99.0%	98.2%
Standard Transformer	90.0%	89.3%	89.9%	84.0%

After completing the training process for the DTrOCR model, it was tested on an additional dataset to evaluate its generalization capabilities and validate the improvements observed during training. When tested on the custom dataset, the Dual Encoder Transformer maintained its high accuracy, as shown in Table 4 and Figure 9.

Furthermore, for the accuracy of the model when dealing with diacritics, it achieved high accuracy, reaching 89% compared to the models that dealt with diacritics before, but less accuracy compared to without diacritics using our generated dataset. Table 5 illustrates the results of DTrOCR when testing it using a dataset with diacritics and without diacritics.

Table 4. Comparison of evaluation metrics for different datasets.

Dataset Used for Test	Accuracy	F1-Score	Precision	Recall
Our Dataset	99.3%	99.1%	99.0%	99.2%
APTI	96.5%	96.1%	96.2%	96.0%
MMAC	95.1%	95%	95.5%	95.2%
IFN/ENIT	87.0%	87.8%	87%	87.5%

Table 5. Comparison of model with and without Tashkeel.

Model	Accuracy	F1-Score	Precision	Recall
With Tashkeel	89.1%	89.0%	88.7%	90.1%
Without Tashkeel	99.3%	99.1%	99.0%	99.2%

5.2 Limitations

Although the model has achieved high performance, we observed that some errors occur in cases involving overlapping or poorly positioned diacritical marks. For instance, when the shadda and fatha are closely placed on a letter such as Seen, the model may misclassify it as Sheen. These specific failure cases highlight the limitations of the current system in distinguishing fine-grained features.

One of the most common errors was the confusion between Sheen and Seen when shadda overlapped with fatha, resulting in an incorrect character interpretation. Similarly, the model frequently misclassified Dal as Thal when the damma was slightly shifted or faint. We also observed consistent difficulty in distinguishing between Kaf and Faa in cases where the kasra was small. In addition, Taa was sometimes confused with Thaa when the sukun was not clearly printed.

Another recurring problem involved stacked diacritics, such as "shadda and kasra" or "shadda and damma", which the model occasionally detected only partially, leading to either missing diacritics or duplicated outputs. Some samples revealed that the model entirely ignored diacritics when multiple marks were close to each other, producing undiacritized text instead. We also noticed alignment errors where diacritics were shifted to the wrong character, particularly in dense handwritten-like fonts.

6. Conclusion and Future Work

This study introduced a deep-learning model, Dual Encoder Transformer (DTrOCR), designed to enhance Arabic Optical Character Recognition (OCR) by recognizing both discretized and non-discretized words. The dual-encoder approach is applied to Arabic word recognition, enhancing the feature-extraction process by utilizing two encoders that collaborate. Before entering the decoder, we implement a merging process for the features extracted from both encoders using two multi-head attention layers. To ensure that the most relevant information is combined and passed on for further processing, this merging step enhances the model’s ability to capture and utilize complementary features, leading to improved recognition accuracy.

The model was trained on the MFSRHRD dataset, which includes both types of words and achieved the following results: 99.3% accuracy for non-diacritized words and 89.9% accuracy for diacritized words, outperforming previous models that struggled with diacritics. To test generalization, the DTrOCR was evaluated on new datasets it had not previously seen and it maintained a strong performance compared to older models, demonstrating its reliability for accurate Arabic-text recognition. In future work, enhancing diacritic recognition remains a crucial challenge.

In future studies, to address the limitations of the model, we propose incorporating specialized attention mechanisms and employing multi-task learning frameworks explicitly designed to capture and differentiate diacritical features. Additionally, exploring model-optimization techniques to reduce computational costs and improve training efficiency will be essential. Lastly, further architectural refinements could facilitate faster training and enable deployment in resource-constrained environments.

References

[1] A. Wasfy et al., "QARI-OCR: High-fidelity Arabic Text Recognition through Multimodal Large

Language Model Adaptation," arXiv preprint, arXiv: 2506.02295, 2025.
[2] A. Waly, B. Tarek, A. Feteha, R. Yehia, G. Amr, W. Gomaa and A. Fares, "Invizo: Arabic Handwritten Document Optical Character Recognition Solution," arXiv preprint, arXiv: 2502.05277, 2025.
[3] A. Chan, A. Mijar, M. Saeed, C.-W. Wong and A. Khater, "HATFormer: Historic Handwritten Arabic Text Recognition with Transformers," arXiv preprint, arXiv: 2410.02179, 2025.
[4] M. Dahbali, N. Aboutabit and N. Lamghari, "A Hybrid Model for Arabic Script Recognition Based on CNN-CBAM and BLSTM," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 10, no. 3, pp. 294–305, DOI: 10.5455/jjcit.71- 1709571516, Sep. 2024.
[5] S. Raminedi, S. Shridevi and D. Won, "Multi-modal Transformer Architecture for Medical Image Analysis and Automated Report Generation," Scientific Reports, vol. 14, no. 1, p. 19281, 2024.
[6] K. Gaashan and M. B. Younes, "Deep Learning-based Arabic Optical Character Recognition: A New Comprehensive Dataset at Character and Word Levels," Proc. of the 2024 15th IEEE Int. Conf. on Information and Communication Systems (ICICS), pp. 1–6, Irbid, Jordan, 2024.
[7] R. Najam and S. Faizullah, "Analysis of Recent Deep Learning Techniques for Arabic Handwritten-text OCR and Post-OCR Correction," Applied Sciences, vol. 13, no. 13, p. 7568, 2023.
[8] A. Mortadi et al., "ALNASIKH: An Arabic OCR System Based on Transformers," Proc. of 2023 IEEE Int. Mobile, Intelligent and Ubiquitous Computing Conf. (MIUCC), pp. 74–81, Cairo, Egypt, 2023.
[9] S. Faizullah, M. S. Ayub, S. Hussain and M. A. Khan, "A Survey of OCR in Arabic Language: Applications, Techniques and Challenges," Applied Sciences, vol. 13, no. 7, p. 4584, 2023.
[10] S. Alghyaline, "Arabic Optical Character Recognition: A Review," Computer Modeling in Engineering & Sciences, vol. 135, no. 3, pp. 1825–1861, 2023.
[11] R. E. Shtaiwi, G. A. Abandah and S. A. Sawalhah, "End-to-End Machine Learning Solution for Recognizing Handwritten Arabic Documents," Proc. of the 2022 13th IEEE Int. Conf. on Information and Communication Systems (ICICS), pp. 180–185, Irbid, Jordan, 2022.
[12] S. Momeni et al., "Arabic Offline Handwritten Text Recognition with Transformers," Research Square, DOI: 10.21203/rs.3.rs-2300065/v1, 2022.
[13] S. K. Jemni et al., "Enhance to Read Better: A Multitask Adversarial Network for Handwritten Document Image Enhancement," Pattern Recognition, vol. 123, p. 108370, 2022.
[14] M. Eltay et al., "Generative Adversarial Network-based Adaptive Data Augmentation for Handwritten Arabic Text Recognition," PeerJ Computer Science, vol. 8, p. e861, 2022.
[15] M. Boualam et al., "Arabic Handwriting Word Recognition Based on Convolutional Recurrent Neural Network," Proc. of the 6th Int. Conf. on Wireless Technologies, Embedded and Intelligent Systems (WITS 2020), pp. 877–885, Springer, 2022.
[16] Y. M. Alwaqfi, M. Mohamad and A. T. Al-Taani, "Generative Adversarial Network for an Improved Arabic Handwritten Characters Recognition," Int. Journal of Advances in Soft Computing & Its Applications, vol. 14, no. 1, pp. 176-195, 2022.
[17] A. Mostafa et al., "OCFormer: A Transformer-based Model for Arabic Handwritten Text Recognition," Proc. of the 2021 IEEE Int. Mobile, Intelligent and Ubiquitous Computing Conference (MIUCC), pp. 182–186, Cairo, Egypt, 2021.
[18] A. T. Al-Taani and S. T. Ahmad, "Recognition of Arabic Handwritten Characters Using Residual Neural Networks," JJCIT, vol. 7, no. 2, pp. 192–205, DOI: 10.5455/jjcit.71-1615204606, Jun. 2021.
[19] R. S. Alkhawaldeh, "Arabic (Indian) Digit Handwritten Recognition Using Recurrent Transfer Deep Architecture," Soft Computing, vol. 25, no. 4, pp. 3131–3141, 2021.
[20] R. Yacouby and D. Axman, "Probabilistic Extension of Precision, Recall and F1-score for More thorough Evaluation of Classification Models," Proc. of the 1st Workshop on Evaluation and Comparison of NLP Systems, pp. 79–91, DOI: 10.18653/v1/2020.eval4nlp-1.9, 2020.
[21] Z. Noubigh et al., "Transfer Learning to Improve Arabic Handwriting Text Recognition," Proc. of the 2020 21st IEEE Int. Arab Conf. on Information Technology (ACIT), IEEE, pp. 1–6, Giza, Egypt, 2020.
[22] M. Fasha, B. Hammo, N. Obeid and J. AlWidian, "A Hybrid Deep Learning Model for Arabic Text Recognition," Int. Journal of Advanced Computer Science and Applications, vol. 11, no. 8, 2020.
[23] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 30, 2017.
[24] G. A. Abandah et al., "Challenges and Pre-processing Recommendations for MADCAT Dataset of Handwritten Arabic Documents," Proc. of the 2018 IEEE 11th Int. Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–9, Beijing, China, 2018.
[25] S. Faizullah et al., "A Survey of OCR in Arabic Language: Applications, Techniques and Challenges," Proc. of the 2015 IEEE Int. Conf. on Communication, Networks and Satellite (COMNESTAT), vol. 13, pp. 111–114, 2015.
[26] S. A. Mahmoud et al., "KHATT: Arabic Offline Handwritten Text Database," Int. Workshop on Frontiers in Handwriting Recognition (IWFHR), Bari, Italy, pp. 449–454, 2012.
[27] A. AbdelRaouf, C. A. Higgins, T. Pridmore and M. Khalil, "Building a Multi-modal Arabic Corpus (MMAC)," Int. Journal on Document Analysis and Recognition, vol. 13, no. 4, pp. 285–302, Dec. 2010.

[28] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi and J. Hennebert, "A New Arabic Printed Text Image Database and Evaluation Protocols," Proc. of the 2009 IEEE 10th Int. Conf. on Document Analysis and Recognition, pp. 946–950, Barcelona, Spain, 2009.

[29] N. Ben Amara et al., "ARABASE: A Relational Database for Arabic OCR Systems," International Arab Journal of Information Technology, vol. 2, pp. 259–266, 2005.

[30] M. Pechwitz et al., "IFN/ENIT-database of Handwritten Arabic Words," Proc. of Francophone Int. Conf. on Writing and Document (CIFED), vol. 2, Citeseer, pp. 127–136, Hammamet, Tunisia, 2002.

ملخص البحث:

تُعدّ المخطوطاتُ باللُّغة العربية من أكثر المخطوطاتِ تعقيداً وصعوبةً؛ فهي تستخدم أشكالاً مختلفة للأحرف مع علامات تشكيلٍ معقّدةٍ من الصّعْب تمييزها عن النّقاط الّتي تحتوي عليها الأحرف المنقوطة. وإنّ الخصائص المميّزة لتلك المخطوطات تجعل التّمييز الضّوئي للأحرف تنطوي على الكثير من التّحدّيات تنجم عنها دِقّة تمييزٍ منخفضة. والجدير بالذكر أنّ الأدبيات تَعُجُّ بالدّراسات الّتي تهدف إلى تقديم أنظمة تمييز ضوئي للأحرف عالية الدّقّة. إلّا أنّ مسألة تحسين الدّقّة في أنظمة التّمييز الضّوئي للأحرف بالعربية تظلّ مسألة مفتوحة تعتمد على مجموعات البيانات المستخدمة ونظام التّمييز المقترح. هنا إضافة إلى ما تضيفه علامات التّشكيل الّتي توضع فوق الأحرف أو تحتها من صعوباتٍ تؤدي إلى انخفاض دقّة تمييز الأحرف.

تقترح هذه الدّراسة نظاماً محسَّناً على مستوى الكلمة للتّمييز الضّوئي للأحرف في مخطوطاتٍ باللّغة العربية يستخدم المحوّلات، إضافة إلى مُرَمِّزيْن اثنيْن. وقد بلغت دقّة التّمييز للنّظام المقترح (98.5%) عند استخدامه في تمييز الأحرف في مخطوطاتٍ تحتوي على نصوصٍ بلا علامات تشكيل، بينما وصلت دقّة التّمييز للنّظام المقترح إلى (89.9%) في المخطوطات الّتي تتضمّن نصوصاً مع علامات تشكيل، ممّا يعني تَفوُّق النّظام المقترح على غيره من الأنظمة المشابهة الواردة في أدبيات الموضوع.