Oblevit at AR-MS NAKBA NLP 2026 Subtask 2: Hybrid CNN–BiLSTM–CTC Framework with Linguistic Refinement for Arabic Handwritten Manuscript Recognition

Reem Akram Juhash, Abuelgasim Sami Abusonoun, Sara Ayad Al-Desouky
Computer Science Department, Arab Open University, Saudi Arabia
22460101ksa@aou.edu.sa, 22451751@aou.edu.sa, s.ayad@arabou.edu.sa

Abstract

Arabic handwritten manuscript recognition remains a challenging problem in Optical Character Recognition (OCR) because of the cursive nature of the Arabic script, variations in handwriting styles, and character overlap. In addition, many Arabic letters are different only in the number or placement of dots, which increases ambiguity in recognition tasks. These challenges require models capable of extracting meaningful visual features while modeling long-range sequential dependencies. In this work, we propose a deep learning-based OCR system for Arabic handwritten text recognition. The proposed approach combines convolutional neural networks (CNN) for feature extraction, bidirectional recurrent neural networks (BiLSTM) for sequence modeling, and Connectionist Temporal Classification (CTC) for end-to-end training without character-level segmentation. To further enhance performance, decoding and specially designed post-processing strategies for Arabic text are applied to handle spacing issues and reduce character-level ambiguity. Experimental results show that the proposed method achieves competitive performance and effectively handles complex handwritten Arabic manuscripts.

Keywords: Optical Character Recognition (OCR), Arabic Handwritten Text Recognition, Deep Learning, CNN–BiLSTM–CTC, Arabic Manuscript Recognition

1. Introduction

Handwritten text recognition remains difficult compared to printed OCR, especially for Arabic script. Arabic handwriting has a cursive nature, letters change shape depending on position, and many characters different from each other only by dots. In historical manuscripts, noise, faint ink, and irregular spacing further increase recognition complexity. Previous systems relied on manual features and deep learning techniques, but modern deep learning methods enable end-to- end training. In this work, we employ the CNN– BiLSTM–CTC framework to jointly model visual and sequential information without explicit segmentation. To improve accuracy, we integrate Arabic-aware decoding and post-processing strategies that address dot ambiguity and word segmentation issues. The proposed system provides a robust solution for recognizing handwritten Arabic manuscripts. The implementation of our system is publicly available to support reproducibility.[1]

2. Related Work

Markov Models (HMMs) were among the earliest approaches used for handwritten text recognition, where handwriting was modeled as a probabilistic sequence problem (Ploetz and Fink, 2009). These systems often relied on handcrafted feature extraction and statistical language models to improve recognition accuracy (Bunke et al., 2009). Although effective for their time, HMM-based methods struggled to capture long-range dependencies and required complex preprocessing pipelines.

The introduction of Connectionist Temporal Classification (CTC) marked a major shift toward end-to-end sequence modeling (Graves et al., 2009). CTC enabled neural networks to be trained without explicit alignment between input images and target text sequences. Graves et al. (2009) demonstrated that combining recurrent neural networks with CTC significantly improved performance in handwriting recognition tasks. Later, multi-dimensional LSTM (MDLSTM) networks were introduced to better capture spatial context in handwritten images (Graves and Schmidhuber, 2008).

With the success of convolutional neural networks in image processing tasks (Krizhevsky et al., 2012), researchers began integrating CNNs with recurrent layers to form hybrid architectures such as CNN–BiLSTM–CTC models. These architectures extract visual features using convolutional layers and then model sequential dependencies using recurrent layers, achieving strong performance in both word-level and line-level handwritten recognition. Attention-based sequence-to-sequence models were later proposed to address some limitations of CTC-based systems (Kass and Vats, 2022; Michael et al., 2019). These models use an encoder–decoder framework with attention mechanisms that dynamically focus on relevant image regions while generating text sequences.

More recently, transformer-based architectures

[1] Source code and trained model are available at https://github.com/aboalgasimm/Nakba-2026.

Proceedings of the 2nd International Workshop on Nakba Narratives as Language Resources (Nakba-NLP 2026) @ LREC 2026, pages 234–238

have been applied to text recognition due to their strong sequence modeling capabilities (Vaswani et al., 2017). Vision Transformers and hybrid CNN–transformer models have demonstrated competitive performance in various recognition tasks (Dosovitskiy et al., 2020). In addition to architectural improvements, data augmentation techniques have been widely used to enhance generalization and robustness (Shorten and Khoshgoftaar, 2019; Wigington et al., 2017). Synthetic data generation has also been explored to overcome limited training data availability (Li et al., 2021). Test-time augmentation strategies have further improved performance by aggregating predictions from multiple transformed versions of the same image (Poznanski and Wolf, 2016).

Building on these developments, our work adopts a CNN–BiLSTM–CTC framework enhanced with tailored decoding and post-processing techniques for Arabic handwritten manuscripts. Unlike purely transformer-based approaches, our model balances performance and computational efficiency while addressing specific challenges related to Arabic script characteristics. We evaluated our system on the AR-MS benchmark introduced in the NakbaNLP 2026 shared task (Hamoud et al., 2026).

3. Dataset

The dataset used in this work was provided as part of the AR-MS shared task introduced in the NakbaNLP 2026 workshop (Hamoud et al., 2026). The training dataset contains thousands of images of Arabic handwritten manuscript lines along with their corresponding text transcriptions as a CSV file. The training dataset was used to train the OCR model and learn the mapping between handwritten images and text sequences. The shared task also provides a blind test dataset that was used for the final evaluation in the leaderboard. It contains up to 2671 images with an empty CSV file that the system must fill after reading and processing the images in order. The challenges of Arabic handwritten manuscripts make the recognition task more difficult compared to printed text. In this work, we only used the official dataset provided by the shared task organizers. No additional external datasets were used.

4. Approach

In this work, we developed a deep learning-based Optical Character Recognition (OCR) system for Arabic handwritten manuscript recognition. The main objective was to design a model capable of handling the unique challenges of Arabic handwriting, including connected characters, dot ambiguity, irregular spacing, and visual noise in historical documents. The system consists of five stages as shown in Figure 1.

4.1. Convolutional Neural Network (CNN)

The first stage of the system is responsible for extracting visual features from the input manuscript images. A convolutional neural network (CNN) is used to transform raw image pixels into higher-level feature representations. The CNN consists of multiple convolutional layers followed by nonlinear activation functions and pooling layers, which progressively capture spatial patterns in handwritten text. CNNs are particularly effective for handwriting recognition because they can detect local visual structures such as edges, strokes, curves, and diacritics. These features are essential for Arabic handwriting since many characters share similar shapes and differ only by the presence or placement of dots. In addition, the CNN helps capture variations in writing style, pen pressure, stroke thickness, and scanning noise. After the convolutional feature extraction stage, the resulting feature maps are reshaped into a sequence representation where the horizontal axis of the text line is treated as a time sequence. This sequential representation is then passed to the recurrent layers for further processing.

4.2. Bidirectional Long Short-Term Memory (BiLSTM)

In the second stage, the extracted CNN features are processed using Bidirectional Long Short-Term Memory (BiLSTM) layers. Recurrent neural networks are well suited for modeling sequential data because they capture dependencies across time steps. The bidirectional structure allows the network to process the sequence in both forward and backward directions. This enables the model to consider both preceding and following context when predicting characters. Such contextual modeling is especially important for Arabic script, where character shapes depend on their position within a word and where correct interpretation often requires surrounding context. The output of the BiLSTM layers is a sequence of context-aware feature vectors that encode both spatial information from the CNN and sequential dependencies from the recurrent network.

4.3. Connectionist Temporal Classification (CTC)

This stage contains the CTC layer, the CTC loss, and the decoding logic. Because we do not have exact alignment between image positions and characters, CTC addresses this problem by producing

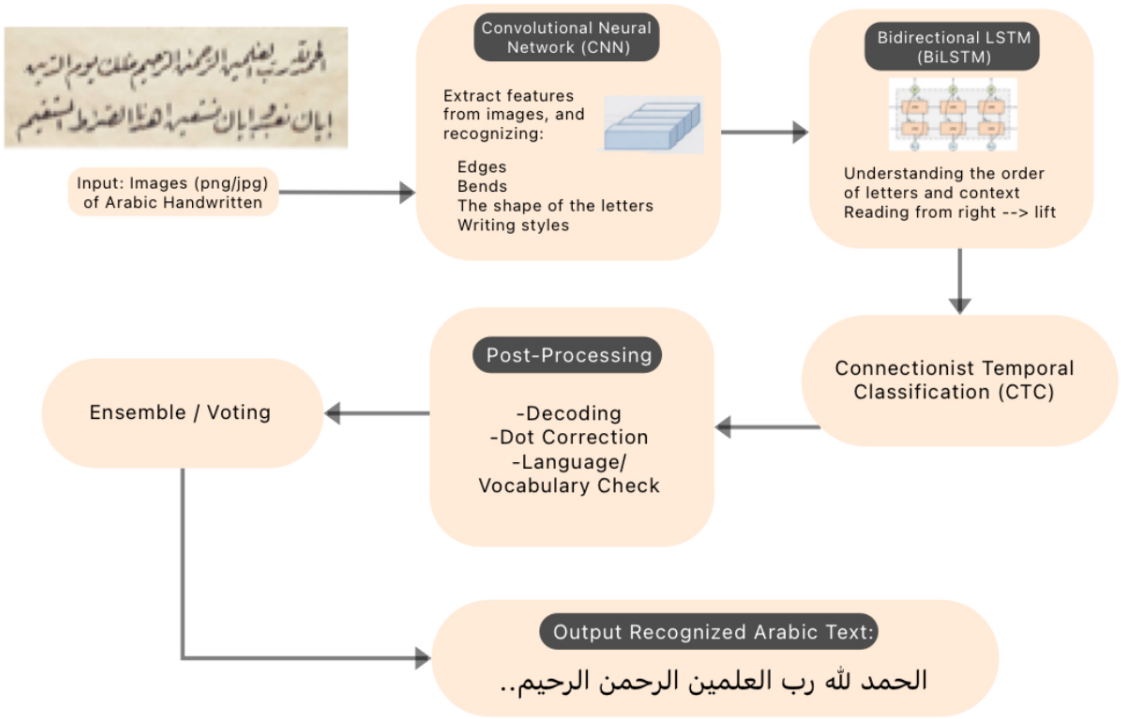

Figure 1: Overview of the proposed Arabic handwritten text recognition framework.

a probability distribution over characters at each time step while allowing blank steps. It learns how to map time steps to character sequences without forced alignment. During decoding, repeated characters are collapsed, blank tokens are removed, and a final character sequence is produced. The output of this stage is a raw predicted text string that is often still noisy and therefore benefits from additional refinement.

4.4. Post-Processing

This stage makes the text cleaner and more correct. It contains several sub-steps. First, beam search decoding is used to explore multiple likely sequences and reduce common CTC errors such as missing or duplicated characters. In our implementation, a beam width of 5 was used and the best sequence was selected based on a combination of CTC probability and lexicon matching. Second, dot correction was applied because Arabic accuracy is heavily affected by dots. We used a rule-based mapping derived from common Arabic character confusions together with context-aware rules. We also applied lexicon-based correction by matching predicted words against a vocabulary extracted from the training dataset, which contains approximately 41,000 unique words. Finally, language and vocabulary checks were used during decoding to prioritize valid Arabic word sequences. Candidates were filtered using this lexicon and the closest valid word was selected using edit distance. No external pretrained language model was used; instead, we relied on lexicon-based constraints.

4.5. Ensemble and Voting Strategy

In our work, we used multiple pipelines, where each pipeline applied different decoding and post-processing methods. Each pipeline was implemented separately and produced its own submission file. After running all experiments, we evaluated the results on Codabench and selected the best-performing outputs. In total, we chose seven files because they consistently achieved the highest results. In the final step, we combined these files using a voting-like approach. All submissions were aligned by filename and compared line by line. We then selected the best prediction for each line based on agreement between candidates and consistency with the training data. This approach helped improve the results by combining the strengths of multiple pipelines instead of relying on a single method.

5. Experimental Setup

Before training, the input images were preprocessed by resizing them to 1536 $\times$ 64 and normalizing pixel values. Since Arabic text is written from right to left, the images were horizontally flipped to preserve the correct reading order. The dataset

was then split into training and validation sets using a 90%–10% split. The model was trained using a CNN–BiLSTM–CTC architecture, where convolutional layers were used to extract visual features, followed by two Bidirectional LSTM layers with 256 and 128 units to capture sequential dependencies between characters. The final prediction layer produced 139 output classes, including the CTC blank token, based on a vocabulary of 138 symbols. Training was performed for 35 epochs with a batch size of 64, using the Adam optimizer with a learning rate of 3×10^{-4}. The CTC loss function was used to enable alignment-free training between input images and target text sequences. Early stopping and learning rate reduction were also applied to improve convergence and prevent overfitting. Performance was evaluated using Character Error Rate (CER) and Word Error Rate (WER). During inference, beam search decoding was applied to generate candidate sequences, followed by multiple post-processing pipelines. Although no full ablation study was conducted, we observed that removing post-processing and lexicon constraints degrades performance, especially in WER. The final system achieved a CER of 0.0925 and a WER of 0.3268.

6. Results and Discussion

The official leaderboard results are shown in Table 1. Our system ranked 2nd among the participating teams with a CER of 0.0925 and a WER of 0.3268. The results show that the proposed approach achieved competitive performance compared to other systems. In contrast, the baseline system obtained a much higher error rate, highlighting the effectiveness of deep learning approaches for Arabic handwritten manuscript recognition.

These results confirm that the CNN-BiLSTM-CTC architecture effectively captures both visual features and sequential dependencies. During experimentation, we faced several challenges. First which was a major issue, dot ambiguity because Arabic letters differ only by dots as we solved in section 4.1. Faint or misplaced dots often led to incorrect character predictions negatively affecting CER. Second, Arabic handwriting has a cursive nature, word merging and splitting errors occurred, which in this case negatively affecting WER however this problem also was solved as mentioned in section 4.1. Third, some noise distortions were misinterpreted as characters. Finally, we noticed some words out-of-vocabulary, some of which were correct rare words, while others resulted from minor letter-level errors. To mitigate these problems, we implemented post-processing techniques as mentioned in the approach above. Overall, the results demonstrate that combining deep learning with targeted decoding and post- processing methods enhances Arabic handwritten text recognition accuracy.

Rank	Username	CER	WER
1	Misraj Ai	0.0790	0.2440
2	Oblevit	0.0925	0.3268
3	3reeq	0.0938	0.2996
4	Latent Narratives	0.1050	0.3106
5	Al-Warraq	0.1142	0.3780
6	Not Gemma	0.1217	0.3063
7	NAMAA-Qari	0.1950	0.5194
8	Fahras	0.2269	0.5223
9	baseline	0.3683	0.6905

Table 1: Official leaderboard results of the ARMS shared task.

7. Conclusion

This paper presented a deep learning OCR system for Arabic handwritten manuscripts using a CNN–BiLSTM–CTC architecture. The model achieved competitive CER and WER results and showed strong ability to handle the complexity of Arabic script. Although challenges such as dot ambiguity and word segmentation errors were observed, tailored post-processing improved robustness.

8. Ethics Statement

This work focuses on Arabic handwritten manuscript recognition using the dataset provided by the shared task organizers. The dataset consists of historical manuscript images and does not contain personal or sensitive information. The purpose of this research is to support the digitization and automatic analysis of historical Arabic documents. The proposed system is intended only for research purposes and aims to improve the accessibility and preservation of historical manuscripts. No personal data was collected or used in this study. We also ensured that the dataset was used according to the guidelines provided by the shared task organizers. The system was not designed for profiling or decision-making about individuals, and its intended use is limited to research and preservation-oriented applications. By improving access to handwritten archival material, the work supports cultural preservation while remaining within the scope defined by the shared task.

9. Bibliographical References

H. Bunke, S. Bengio, and A. Vinciarelli. 2009. Offline recognition of unconstrained handwritten texts using hmms and statistical language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, and D. Weissenborn. 2020. An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. 2009. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868.

A. Graves and J. Schmidhuber. 2008. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in Neural Information Processing Systems*.

H. Hamoud, A. A. Chamseddine, B. Shalash, F. B. Abid, M. Jarrar, C. A. Chakra, B. Ghanem, and F. A. Zaraket. 2026. Nakba nlp 2026: Shared task on arabic handwritten manuscript understanding (palestine memory—omar al-saleh memoir). In *Proceedings of the 2nd International Workshop on Nakba Narratives as Language Resources (Nakba-NLP 2026), co-located with LREC 2026, Palma, Mallorca, Spain*.

D. Kass and E. Vats. 2022. Attentionhtr: Handwritten text recognition based on attention encoder–decoder networks. In *International Conference on Document Analysis Systems*.

A. Krizhevsky, I. Sutskever, and G. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.

M. Li, T. Lv, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei. 2021. Trocr: Transformer-based optical character recognition with pre-trained models. *arXiv preprint arXiv:2109.10282*.

J. Michael, R. Labahn, T. Gruning, and J. Zollner. 2019. Evaluating sequence-to-sequence models for handwritten text recognition. In *International Conference on Document Analysis and Recognition*, pages 1286–1293.

T. Ploetz and G. Fink. 2009. Markov models for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 12:269–298.

A. Poznanski and L. Wolf. 2016. Cnn-n-gram for handwriting word recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2305–2314.

C. Shorten and T. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1).

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

C. Wigington, S. Stewart, B. Davis, B. Barrett, B. Price, and S. Cohen. 2017. Data augmentation for recognition of handwritten words and lines using a cnn–lstm network. In *International Conference on Document Analysis and Recognition*, pages 639–645.