Neural Computing and Applications (2022) 34:2055–2071
https://doi.org/10.1007/s00521-021-06520-7

ORIGINAL ARTICLE

Domain and writer adaptation of offline Arabic handwriting recognition using deep neural networks

Sana Khamekhem Jemni[1,2] · Sourour Ammar[1,3] · Yousri Kessentini[1,3]

Received: 3 November 2020 / Accepted: 7 September 2021 / Published online: 21 September 2021

Abstract
Arabic Handwritten Text Recognition (AHTR) based on deep learning approaches remains a challenging problem due to the inevitable domain shift like the variability among writers' styles and the scarcity of labelled data. To alleviate such problems, we investigate in this paper different domain adaptation strategies of AHTR system. The main idea is to exploit the knowledge of a handwriting source domain and to transfer this knowledge to another domain where only few labelled data are available. Different writer-dependent and writer-independent domain adaptation strategies are explored using a convolutional neural networks (CNN) and Bidirectional Long Short Term Memory (BSTM) - connectionist temporal classification (CTC) architecture. To discuss the interest of the proposed techniques on the target domain, we have conducted extensive experiments using three Arabic handwritten text datasets, mainly, the MADCAT, the AHTID/MW and the IFN/ENIT. Concurrently, the Arabic handwritten text dataset KHATT was used as the source domain. The obtained results prove the effectiveness of the proposed strategies specially when considering the writer's information during the supervised adaptation process.

Keywords Domain adaptation · Writer adaptation · Arabic offline handwriting recognition · CNN · BLSTM

1 Introduction

The recognition of Arabic handwritten text remains an open issue in the pattern recognition field. Massive amount of archived documents are waiting to be processed. The production of electronic versions of these documents is required in order to avoid exhaustive and time-consuming of manual search for information and to retrieve the required request easily using text queries. Handwriting text recognition (HTR) has been successfully used in many applications such as bank check reading, postal codes recognition, mail sorting, book transcription, document analysis and many other related tasks. Although, the recognition of offline Arabic handwritten script is still challenging because of the writing variability among writers and the scarcity of labelled data.

Traditionally, the common approach used for the design of a HTR system was mainly based on Hidden Markov Models (HMMs) [1–3]. However, standard HMMs present a major problem when the task involves long-term dependencies. To overcome this issue, recurrent neural networks (RNN), in particular Long-Short Term Memory (LSTM), have been successfully applied in the field of cursive handwriting recognition. LSTMs have been widely used for speech recognition [4], text recognition [5], emotion recognition [6] and time series prediction problems [7] thanks to their ability of sequence learning. The LSTM architecture has given the RNN a rebirth by overcoming many shortcomings of earlier RNN architectures [8]. It was emerged as a competent classifier for handwriting recognition tasks since it performs considerably

✉ Sana Khamekhem Jemni
sana.khamekhemjemni@enis.tn

Sourour Ammar
sourour.ammar@crns.rnrt.tn

Yousri Kessentini
yousri.kessentini@crns.rnrt.tn

[1] Digital Research Center of Sfax, B.P. 275 Sakiet Ezzit, 3021 Sfax, Tunisia

[2] MIR@CL: Multimedia, InfoRmation systems and Advanced Computing Laboratory, Sfax, Tunisia

[3] SM@RTS : Laboratory of Signals, systeMs, aRtificial Intelligence and neTworkS, Sfax, Tunisia

well on handwritten text without an explicit language's knowledge. In order to access to the context in both forward and backward directions, the author in [9] introduced the Bi-directional LSTM (BLSTM) architecture. The BLSTM is a combination of bi-directional RNN and LSTM, and it uses two hidden layers, one for forward pass and the other for backward pass. Since pre-segmented input data are a peculiar constraint of the original RNNs, the authors in [10] used a forward-backward algorithm to align transcriptions with the output of the neural network that is referred to as connectionist temporal classification (CTC). The BLSTM-CTC architectures have shown greater recognition accuracy on Arabic handwriting text recognition task [11–15] throw their capacity to model long sequential data. The BLSTM networks extension to the Multi-Dimensional LSTM (MDLSTM) has been successfully applied in handwriting recognition allowing to build systems that can handle both the 2D aspect of an input image and the sequential aspect of the prediction. Particularly, the MDLSTM combined with the CTC objective function, have brought to a low error rate and became the state-of-the-art model for handwriting recognition systems [14–19]. Recent works on Convolutional Neural Network (CNN) architectures [20] [19, 21–23] showed that they are competitive with RNN architectures even for modelling long-term dependencies. More recently, several research works have focused on improving deep learning architectures through the inclusion of an attention mechanism in order to focus on the most relevant features of the input data. Similar to a human who inspects a scene through eye movements, the attention mechanism allows a neural network to focus on a relevant part of the input data [24]: part of a face for facial recognition [25] or part of a text line image for handwriting recognition [26]. Many researchers have been motivated by the concept of the attention mechanism and have applied it to the recognition of single words and handwritten text lines [26–30]. For instance, authors in [30] investigated the use of encoder-decoder networks based on an attention mechanism. Given a handwritten text image, the proposed system [30] extracts relevant features using stacked CNNs. The encoded image is then decoded using RNNs including an attention mechanism. Interesting results were obtained without using neither lexicons nor statistical language models. It has been demonstrated also that systems based on an attention mechanism [30] lead to comparable recognition results with those based on a CNN-BLSTM-CTC architecture [29] or a MDLSTM-CTC model [22].

As outlined above, deep learning approaches have acquired greater importance in the field of handwriting text recognition leading to better results across the use of large amounts of labelled data. However, it remains numerous issues to be resolved, especially, when labelled data used for a given task are too costly to collect, annotate, or even both. Therefore, a simple deep learning approach would not be efficient in such context. As a result, it becomes difficult to build a handwritten text recognition system that can effectively handle new scenarios with limited and fluctuating data. This problem is usually caused by the inherent domain shift when real data are encountered in the testing phase. The data distribution tends to be altered by multiple factors such as the variation of handwriting styles when dealing with new handwritten documents. Indeed, training a classifier on a specific dataset and testing it on another one yields non-optimal results due to the effect of covariate shift [31] and the dataset bias [32]. This covariance is induced by the difference in the image's distributions between datasets. Prior works tried to minimize this shift differently. Some methods are evaluated on handwritten text in natural scene and on character datasets. These methods tried to recognize letters and they have shown an effective performance. Recent works on domain adaptation are based on deep CNN architectures [33] to align the source and target domains into a shared space. These methods attempted to improve the overall representation by minimizing the domain shift [33, 34]. Others used a function to map features from the source domain to the target one [35–37]. For instance, authors in [37] proposed a Style Transfer Mapping (STM) method, based on an affine transformation, to project the writer's data into a style-free space. The proposed STM method has led to a significant improvement on the recognition of a target dataset consisting of handwritten Chinese characters. The used affine transformation allowed to reduce the style variability among writers. Although their effectiveness, these methods cannot be immediately applied to images of handwritten lines of text, because lines are sequential data including several differently written characters.

Despite the considerable efforts supplied on handwriting recognition using advanced deep learning techniques, the domain adaptation of Arabic HTR systems based on a CNN-BLSTM architecture remains unexploited. Therefore, we focus in this work on developing a domain adaptation strategy dealing with the scarcity of labelled data for a HTR task and solving domains shift problems. In this context, we exploit the knowledge of a handwriting source domain, and we transfer this latter to another domain where only a few labelled data are available. Our strategy is able to recognize unseen handwritten characters without any lexicon or language models. We explore, in this work, different adaptation techniques including *writer-dependent* and *writer-independent* to prove the efficiency of incorporating writer's features during the adaptation process. To the best of our knowledge, this is the first study that deals with domain/writer adaptation of a CNN-BLSTM architecture for Arabic handwriting recognition task.

We summarize our contributions as follows:

- We propose a domain adaptation process of a deep learning model to overcome the domains shift issue and writer's styles variation of an Arabic handwritten text recognition system, especially, when few labelled data are available in the target domain.
- The generic HTR system is based on a CNN-BLSTM-CTC architecture and is firstly trained using a large source dataset, namely, the KHATT dataset. Then, the learned knowledge is transferred to the target domain using a two-stage transfer learning (TL). This process allows the extraction of the most relevant features modelling the handwriting styles variation of the target dataset.
- We conducted an extensive evaluation of different domain adaptation strategies, that exploit the writer's information of the target samples, including the global-writer adaptation and the specific-writer adaptation. We also compared these methods to an independent-writer adaptation strategy, where the writer's information is not considered.
- The evaluation of the proposed adaptation strategies is performed using three target datasets, namely, MADCAT, AHTID/MW and IFN/ENIT. The obtained results proved the effectiveness of our proposed approach to reduce the domain shift of heterogeneous datasets.

The rest of this paper is organized as follows. Section 2 provides a review of prior works on domain and writer adaptation techniques. Section 3 presents the architecture of the CNN-BLSTM-CTC model used in this study. Section 4 describes the proposed domain adaptation strategies including writer dependent/independent approaches. The used datasets are described in Section 5. Section 6 presents the experimental settings and discusses the obtained results. Finally, Sect. 7 concludes this paper.

2 Related work

Speech and Handwriting recognition are influenced by the high variability between both speakers and writers. This variability is already confirmed to restrict the good recognition performance of multi-writer/multi-speaker recognition systems. Following the speech recognition community, writer adaptation techniques have been developed to modify early handwritten text recognition models based on Hidden Markov Models (HMMs) [38, 39]. In the case of an Omni-writer HTR system based on HMM model, the model parameters, comprising the means and variances of the Gaussian mixture, can be adjusted to better correlate to the target data distribution. Other works have outlined a strategy for maximizing expectations [40] using a set of distinct character recognizers. These methods present the major advantage of adapting the data to the target domain using an unsupervised manner without the requirement of a labelled set from the target dataset. However, these methods are not suitable for deep learning-based approaches.

As stated before, with the growth of deep learning techniques, the use of a LSTM-based HTR system is becoming more widespread. These data hungry models (on term of labelled data) have been commonly trained using a large dataset and then they are adapted according to the target dataset to be recognized. Usually, in these cases, researchers tend to reuse a model trained from a closely correlated dataset with a large number of samples, known here as the source domain, and then they perform a fine-tuning using the much smaller dataset of interest, known here as the target domain. For example, authors in [41] proposed a system that were trained on a generic dataset, and then fine-tuned on a particular dataset using transfer learning and data augmentation techniques. A more recent work was proposed in [42] where authors attempted to perform the training stage using synthetic data, to help the model to adapt to new styles. But synthetic data can hardly imitate all the particular styles of writers encountered in the real world, especially when the style is very special. Similarly, another recent work was presented in [43] where authors introduced a few-shot learning approach for historical Ciphered manuscript recognition. They performed the training stage using synthetic data and then fine-tuned the model using few real labelled data. These supervised domain adaptation techniques [41, 42, 44] ensure that neural networks can be appropriately trained, eventually by extracting the relevant features from the writing strokes, which are then restructured in the target dataset. However, regardless the efficiency of the above techniques, they are generally writer-independent, where the writer's information is not included in the adaptation process. The incorporation of such information can lead to a significant improvement in the recognition results of the target dataset (test set), as proposed in our present work.

In order to overcome the need of labelled target data, unsupervised domain adaptation techniques have been introduced in the literature [40, 45, 46]. Given a labelled source dataset and an unlabelled target one, the main focus is to adapt the recognition model in order to generalize it to the target domain as well as considering the domain shift between the source and the target datasets. For instance, authors in [45] proposed a domain adaptation strategy designed for Chinese character dataset. Their strategy is based on STM transformation and it has been applied successfully when the source and the target domains are sharing the same character set. Although its efficiency, this method can be useful only for isolated characters and not for sequential data. In fact, such a strategy can accurately

describe characters locally but when dealing with sequential data, such as handwritten text lines, this kind of approaches is not adapted. A typical approach addressing unsupervised domain adaptation is based on adversarial learning [46], in which the disparity between the different domains is reduced through the joint training of a recognition network and a domain discriminator model. For example, authors in [46] presented an unsupervised writer adaptation approach that can automatically adapt a generic handwritten word recognizer, fully trained using synthetic fonts, to a new writer. The proposed HTR system in [46] is based on an encoder-decoder architecture with an attention mechanism. It provides a practical and generic approach dealing with new documents, averting time-consuming and the expense of manual annotation. Despite the interesting results achieved in [46], the proposed system allows only the use of the same character set present in the source dataset. Such limitation prevents the HTR system from recognizing characters in a target dataset containing new letters. Moreover, the proposed approach is restricted to operate at word level and not at line level. In contrast to their approach, we propose in this work a domain adaptation strategy that is able to deal with both handwritten words and text line images.

To summarize, deep learning-based methods are now significantly outperforming the classic handwriting recognition algorithms, based on HMMs, for the handwriting text recognition and domain adaptation tasks. However, the strong dependence on the same character set present in the source/target datasets is limiting severely their performance. Besides, the exclusion of the writer knowledge embedded in the target dataset reduces significantly the performance of the text recognition systems since the writer style particularities are generally neglected.

3 Model architecture

In general, accurate AHTR systems are based on CNN-MDLSTM architecture [15] to model handwritten Arabic text lines. However, the use of the MDLSTM layers at a first stage has some shortcomings meanwhile such architectures are requiring an expensive computational and memory cost. Recently, researchers in [47] have confirmed that the use of only convolutional layers for feature extraction leads to visually similar features. Therefore, we rely in our work on a CNN-BLSTM model for the design of the generic AHTR system. Figure 1 shows the details of the used architecture. We implement the BLSTM network at a higher level in the system structure, and we use five convolutional layers for feature extraction. Additionally, we apply five filters whose sizes are 16, 32, 48, 64 and 80, respectively. At this stage, max-pooling operations are

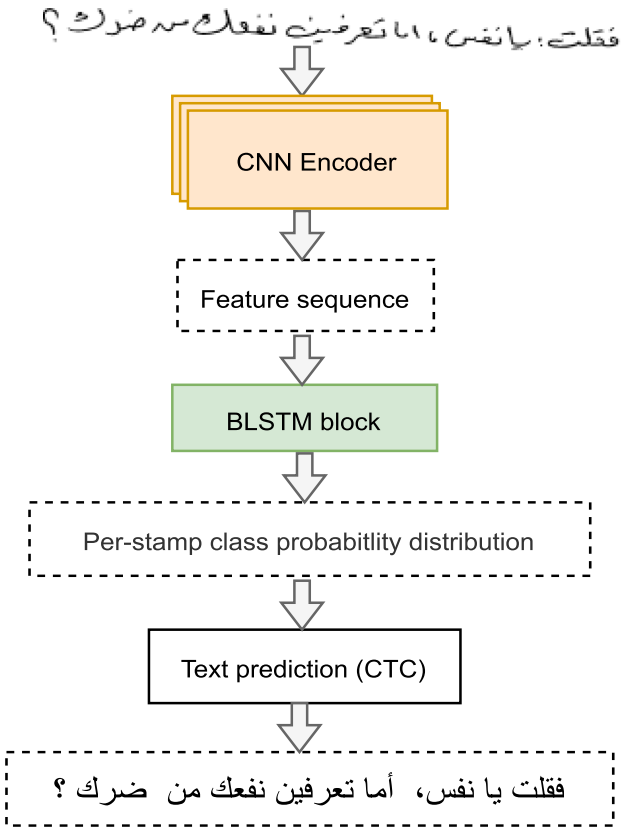

Fig. 1 The architecture of the CNN-BLSTM-CTC model used in this study

applied to reduce the size of the input sequence. The extracted features are processed using two BLSTM recurrent layers of 256 units. In this way, we have 256 features for each direction. A depth-wise concatenation is carried out to adapt the input of the subsequent layer, in overall size of 512. Dropout regularization technique is performed for every layer's output excluding the first CNN one.

To end with, output layers rely on the CTC method [5]. This technique involves a fully connected (FC) layer stacked after the second BLSTM layer to compute the probability distribution for each step throughout the input sequence. It is to note also that the CTC loss function models the conditional label sequences probability regarding the probability distribution of each predicted label (characters). Based on this distribution, an objective function is derived to maximize the probabilities of the correct label sequences. At each time step, the network yields to a probability distribution for the label set $L^{'} = L \bigcup \{blank\}$, where L includes all the character labels in the dataset and the extra blank symbol represents 'no label'. In this work, we use 110 target labels incremented by one extra blank symbol to represent a non-output. The activation y_k^t is defined as the probability of observing a label K of $L^{'}$ at time t. For an input sequence X of length T,

the conditional probability $P(\Pi|X)$ of observing a path Π through the set of label observations is defined by Eq. 1.

$$\mathcal{P}(\Pi|X) = \prod_{t=1}^{T} y_{\Pi_t}^{t} \quad ; \quad \Pi \in L^{'T} \tag{1}$$

Where Π_t represents the observed label at a time t over the path Π, and $L^{'T}$ defines the set of length T paths along $L^{'}$. Paths are mapped to label sequence using an operation B that removes the repeated labels and the blank from the predicted sequence. For a given label sequence l, $l \in L^{U}$, $U \leq T$, Π is more than one corresponds to it. The conditional probability of l is evaluated as the sum of probabilities of all the corresponding paths (Eq. 2).

$$\mathcal{P}(l|X) = \sum_{\Pi \in B^{-1}(l)} P(\Pi|X) \tag{2}$$

Since the above sum computation is considered naive and even unfeasible, it is accurately estimated using the dynamic-programming algorithm. The CTC loss function for a given label sequence is defined as the negative log probability of correctly labelling sequence (Eq. 3).

$$\mathcal{CTC}(l, X) = -\ln(P(l|X)) \tag{3}$$

Where l and X denote the label sequence from ground truth and the output of the BLSTM's final layer, respectively. P indicates the conditional probability of label sequences given the probability distribution of each predicted label.

4 Domain and writer adaptation For Arabic handwriting text recognition

In this work, we propose to evaluate different domain adaptation strategies using a deep CNN-BLSTM-CTC architecture. Specifically, the source domain consists of a large Arabic handwritten text line images with well-annotated text labels. In the target domain, we only have access to few labelled text images including different writing styles. As shown in Fig. 2, the handwriting style of the source domain (here KHATT dataset) is very different of the handwriting style of the target domain (here MADCAT dataset). So, it is challenging to build a handwriting recognition system that handles both the gap between different domains as well as the variability of writer's styles, especially in the case where we have limited size of annotated data.

To build a robust text recognizer for the target domain, the generic optical model, trained on a large dataset (source domain), serves as a starting point. We propose different adaptation strategies to exploit the knowledge of a handwriting source domain and to transfer this knowledge to the target domain. The first strategy, referred to by *Writer Independent Adaptation (WIA)*, is writer-independent and does not integrate any writer's styles information on the adaptation process. The second and the third adaptation strategies are writer-dependent and try to consider the writer's style in the adaptation process. The flow chart of the suggested adaptation strategies is shown in Fig. 3. We present in the next sections the details of each strategy.

4.1 Writer independent adapation (WIA)

Our AHTR system is initially trained on a large Arabic lines images dataset designed by source dataset. As stated earlier, the major crucial issue is that the handwritten images of the target dataset are scanned using different devices and subjected to different noise and writing conditions. Generally, the common adaptation technique used for such task relies on the adjustment of the generic optical model weights by freezing some layers and retraining the others [41]. Giving pixel values of an image as input, the artificial neurons in the CNN can capture a variety of visual features. In this way, the activation maps highlight the relevant features of the image. Usually, the bottom layers of the CNN detect low-level features like horizontal and vertical edges. The output of the first layer is fed into the next layer, which extracts more complex features such as corners. The deeper the CNN architecture, the more meaningful detected features. This is explained by the fact that the latest layers start detecting higher-level features.

Inspired by the work presented in [48], we propose in this study a two-stage adaptation process to transfer meaningful features from the source domain to the target domain in order to take benefits of the large labelled dataset of the source domain (Fig. 4). In the first stage, only the weights of middle layers (Conv 3 and Conv 4) are adjusted. In this way, the remaining layers (Conv 1, Conv 2, Conv 5, BLSTM 1, BLSTM 2 and the linear layer) are frozen.

The motivation of training middle layers in a first stage is inspired from humans. In fact, the human visual system recognizes objects starting from high-frequency

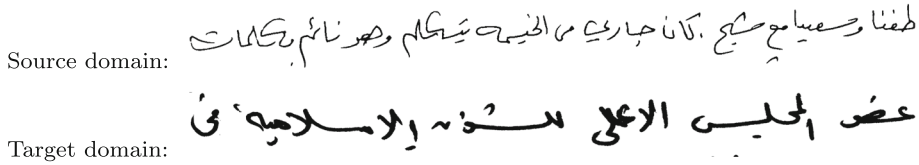

Fig. 2 Examples of handwritten text lines images from source and target domains illustrating the domain shift problem (handwriting style)

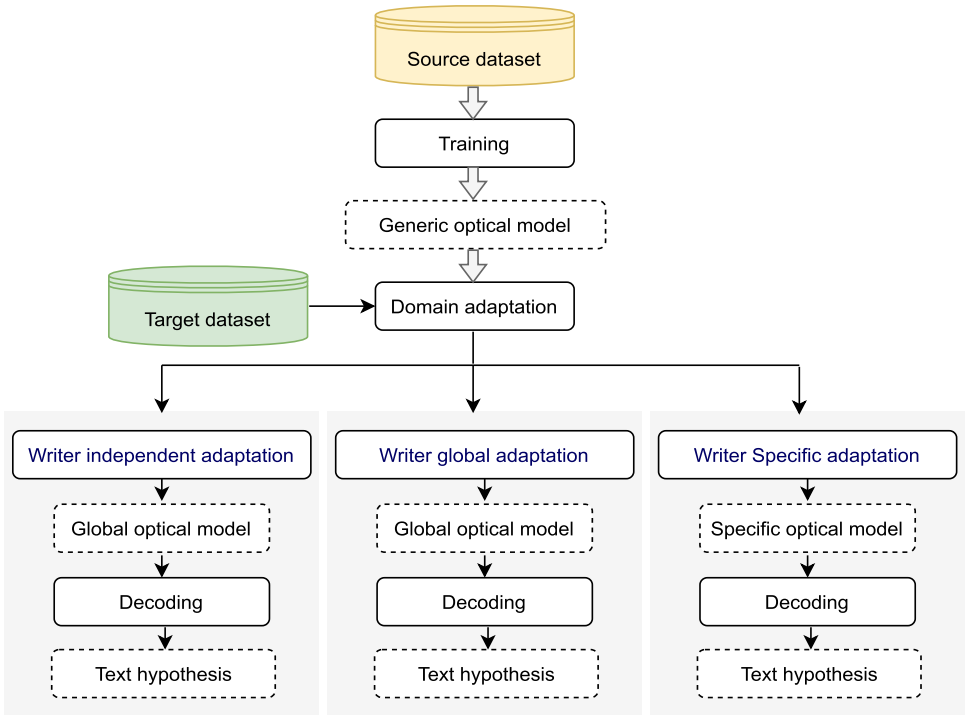

Fig. 3 Overview of the proposed domain adaptation strategies

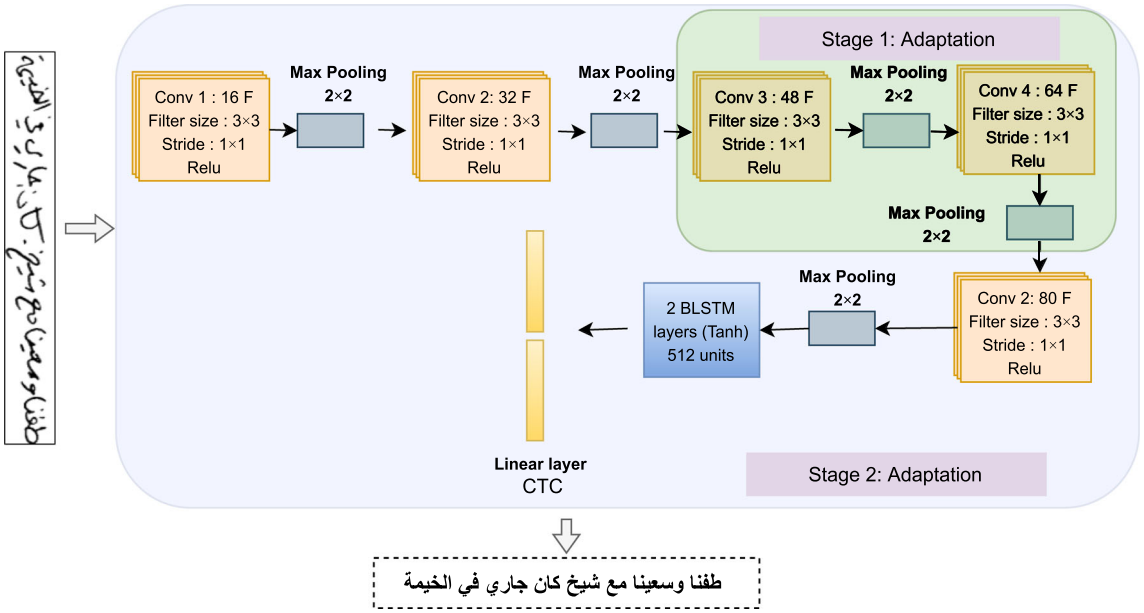

Fig. 4 The used domain adaptation technique: The network is trained in two stages. In stage 1, the parameters for all the layers, expect Convolutional layers 3 and 4 and the final ones marked in blue, are frozen and the system is trained until there is no further improvement in terms of loss. Afterwards, in stage 2, the parameters marked in green, as well as all the first stage's layers, are adjusted during the adaptation

components in the images to high-level object shapes. The edge is a representation of high-frequency, which can be detected in the first layers of a CNN as a low-level feature. It is less known than what happens in the intermediate layers of the human neural network. Mid-level features reflect a blend of low-level and high-level features that may be relevant in identifying intra-class variation. It is to note that the use of intermediate-level features of a trained convolutional neural network, which is a model of a human vision system, has become a trend in the last decade [4, 49]. Considering the relevance of middle layers to extract significant features, we extract mid-level features

by training Conv 3 and Conv 4 intermediate layers in a first stage. However, after several epochs, the learning process saturates. This saturation is defined by the reduction in the objective loss function. So, the first adaptation stage is stopped when no further improvement in terms of CTC loss [50] is noticed. Then, in a second stage, the locked part of the CNN is enabled, which allows the parameters of the other convolutional layers to be adjusted. So that, the BLSTM layer's parameters are adjusted in a manner to select the best-learned features from the final layer of the CNN (Conv 5). In this way, the network's weights are updated by considering the target dataset particularities. Specifically, the network learns handwritten style's variability among writers such us strokes, curves, edges, etc. using a combined mid-level and high-level extracted features.

4.2 Writer dependent adaptation

The adaptation method described above (Sect. 4.1) does not take into consideration the writer's knowledge. It should be noted that the handwriting style variation across writers makes handwriting recognition a challenging problem. In fact, writing styles vary from one writer to another according to the age, the cultural level, the civilization, etc. To address this issue, considering the style particularities of writers in the target dataset could be useful for a better recognition performance. In this way, the system can adapt towards a new handwritten style with the assistance of some writer dependent data from the target dataset. For this reason, we integrate the writer's information into the adaptation process in order to further enhance the WIA recognition results. Here, the dependent writer adaptation relies on learning some additional features from a given writer's samples. Therefore, as a part of the writer adaptation, which depends on the writer's knowledge, we propose to evaluate two writer adaptation strategies to adapt the generic optical model to specific writer's styles, namely, the *Writer Global Adaptation (WGA)* and the *Writer Specific Adaptation (WSA)*.

4.2.1 Writer Global Adaptation (WGA)

The WGA strategy consists of adapting the generic optical model using all writer's features that are incorporated in the test dataset (target domain). So, using the WGA, a unique trained model is then generated based on samples related to all writers figuring in the test set. Assuming that we proceed from a source dataset containing N writers in training and M writers in the target dataset, we design a single model by adapting the model learned on the source domain (N writers) to the M writers' samples (Fig. 5).

4.2.2 Writer specific adaptation (WSA)

The WSA depends also on writer's information styles used in the target dataset. However, the main difference between the WGA and the WSA is that the pre-trained model is adapted according to the specifications of each writer separately. Using the WSA, we train a model for each writer in the target dataset using only his related samples in the adaptation dataset. So, for the WSA task, considering M writers in the target dataset, we adapt M models for the M writers as detailed in Fig. 6.

5 Arabic handwritten datasets

In this section, a detailed description of the used datasets is provided. In this work, we focus on the use of four datasets, namely, the KHATT [51] dataset, the MADCAT dataset [52], the AHTID/MW dataset [53] and the IFN/ENIT dataset [54] consisting of offline Arabic handwritten texts lines. We notice that the four datasets are written by different writers having different ages and different civilizations. That is the main motivation for our evaluation of different domain adaptation strategies for handwriting recognition regardless of the writing style. Fig. 7 presents different text images extracted from the four datasets used in our experiments. This Figure shows the variability in Arabic handwriting text among the different datasets.

5.1 KHATT dataset

The KHATT dataset [51] is a freely off-line handwritten text dataset consisting of 4000 paragraphs written by 1000 distinct writers from different countries, age groups, gender and education level. This dataset contains unrestricted writing styles. In this work, the experiments were conducted on 4000 paragraphs with all the text content and images scanned at 300 dpi resolution. This dataset includes training, development (dev) and test splits. Statistics describing the KHATT dataset sets are shown in Table 1.

5.2 MADCAT dataset

The MADCAT [52] dataset is provided by the MADCAT program within the context of the OpenHaRT evaluation. The data consist of more than 40k handwritten pages with text chosen from web forums and newspapers. In this work, we only use a set of 9000 lines related to 15 writers in order to evaluate the different writer adaptation strategies. A set of 4500 line images are used for the adaptation process, 2250 for validation and the remaining are used for testing. It should be noted that we used 2250 lines for testing,

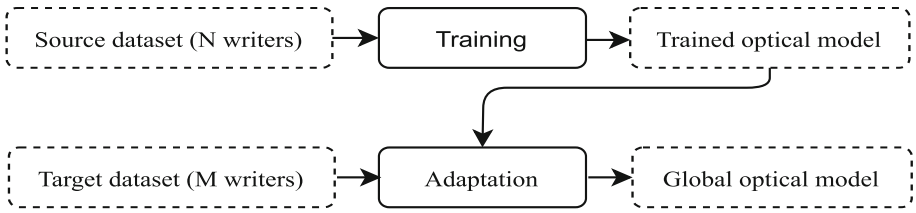

Fig. 5 The proposed Writer Global Adaptation strategy

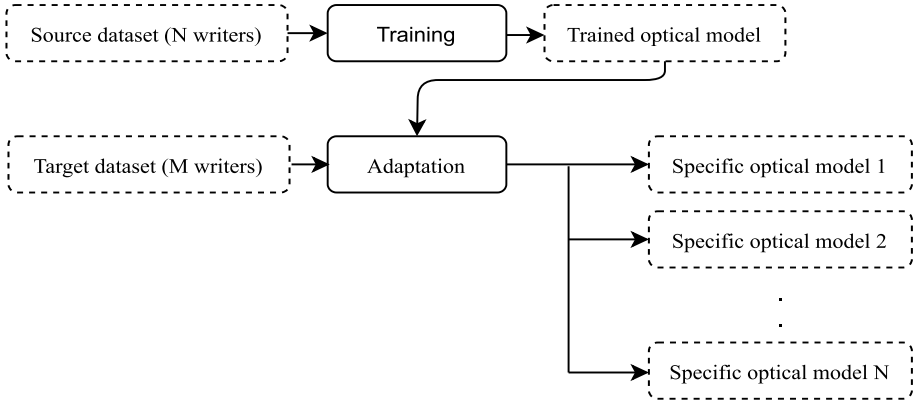

Fig. 6 The proposed Writer Specific Adaptation strategy

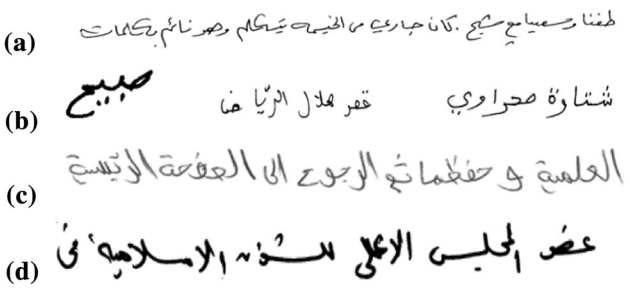

Fig. 7 Sample images randomly selected from different offline Arabic handwritten datasets: **a** KHATT (source domain), **b** IFN/ENIT (target domain), **c** AHTID/MW (target domain), **d** MADCAT (target domain)

Table 1 Statistics of the KHATT dataset

	Train	Test	Dev
Pages	690	141	148
Lines	9475	2007	1902

corresponding to 150 samples per writer. The total number of writers in the test set is equal to 15.

5.3 AHTID/MW dataset

The AHTID/MW handwritten texts lines dataset [53] includes 3710 line images written by 53 writers. The handwritten texts were scanned in grayscale level with the resolution of 300 dpi. The AHTID/MW includes 126 511 characters and 22 896 words. In our experiments, we randomly choose a set of 900 lines to evaluate the proposed domain adaptation strategies. A set of 600 lines are used for the adaptation process. For testing, we used 300 lines, which are corresponding to 20 lines written by each writer. In the test set, the overall writer's number is set to 15.

5.4 IFN/ENIT dataset

The IFN/ENIT [54] dataset is composed of 32 492 images of Tunisian cities and villages names written by 1000 different writers (Table 2). It should be noted that the number of images per writer is approximately of 60 to 90 samples. These images correspond to a vocabulary of 937 words. This dataset is divided into five sets, a, b, c, d and e. In the following experiments, we randomly chosen a set of 1260 word images to evaluate the adaptation strategies. We used 960 for the adaptation, which are corresponding to 64 images written by 15 writers. For testing, we used 300 images corresponding to 20 images written by 15 writers.

Table 2 Statistics of the IFN/ENIT dataset.

Sets	Words	Characters
a	6537	51984
b	6710	53862
c	6477	52155
d	6735	54166
e	6033	45169
Total	32492	257336

6 Experimental results

We present in this section an extensive experimental evaluation of the different domain and writer adaptation strategies. The system's performance is evaluated using the standard Character Error Rate (CER) metric. To evaluate the accuracy of the presented systems, we use the Levenshtein edit distance [55] between the output text and the ground-truth one. Edit distance is calculated by computing the number of edit operations (insertions, substitutions and deletions) that are needed to transform a source string into the target string. The formula of the CER is given below:

$$CER = \frac{dc + sc + ic}{Nc} \times 100 \quad (4)$$

Where dc is the number of deleted characters, sc is the number of substituted characters, ic is the number of inserted characters and Nc is the total number of characters in the reference.

6.1 Implementation details

All our experiments were implemented using Tensorflow on a cluster of NVIDIA GPUs. The training was performed with the Adam optimizer [56] using a learning rate of 0.0003 and a batch size of 16. We have set the dropout value to 0.2 for all the CNN layers except the first layer of the CNN. The training was stopped if there is no improvement on the character error rate value during 20 epochs. Hyper-parameters used in our experiments are detailed in Table 3.

Table 3 Detailed structure of our CNN model

Operations
Conv2d (3×3, 16; stride: 1×1)
Max pooling (2×2; stride: 1×1)
Relu
Dropout (0.0)
Conv2d (3×3, 32; stride: 1×1)
Max pooling (2×2; stride: 1×1)
Relu
Dropout (0.2)
Conv2d (3×3, 48; stride: 1×1)
Max pooling (2×2; stride: 1×1)
Relu
Dropout (0.2)
Conv2d (3×3, 64; stride: 1×1)
Max pooling (2×2; stride: 1×1)
Relu
Dropout (0.2)
Conv2d (3×3, 80; stride: 1×1)
Max pooling (2×2; stride: 1×1)
Relu
Dropout (0.2)

6.2 Evaluation of the two-stage adaptation strategy

Before evaluating the performance of the proposed writer adaptation strategies, we first validate the relevance of the two-stage adaptation process which tries to adapt the optical model trained on a source domain using few labelled data from the target domain. For this purpose, we have trained the generic CNN-BLSTM model using the KHATT dataset (considered as a source domain). We have used 5K line samples for training and a set of 500 images for validation.

To compare the one-stage and the two-stage adaptation techniques, three datasets are used as a target domain (MADCAT, AHTID/MW and IFN/ENIT). For each target dataset, two experiments are conducted depending of the number of adaptation samples (150 samples and 300 samples). Here, it should be noted that this evaluation is performed independently of the writer's information and the obtained results are reported without the integration of any lexicon nor language model.

As shown in Table 4, the two-stage adaptation strategy performs better compared to the one-stage technique using the three target datasets. An average improvement of 1.08 % is obtained using a set of 150 adaptation samples. By augmenting the size of the adaptation samples to 300 images, an average improvement of 1.49 % is achieved. In our following experiments, the two-stage adaptation technique is adopted.

6.3 Evaluation of the writer adaptation strategies

We focus in this section on the benchmarking of the proposed writer adaptation strategies, namely, the writer independent adaptation (WIA), the writer global adaptation (WGA) and the writer specific adaptation (WSA). An Extensive experimental evaluation is conducted using three target datasets with and without the integration of a data augmentation process.

6.3.1 Evaluation on the MADCAT dataset

Table 5 reports the experimental results of our proposed writer adaptation strategies carried out on the MADCAT dataset. As shown in Table 5, a high CER is obtained (42.12%) when no domain adaptation is performed (i.e. the

Table 4 Evaluation of the one-stage and the two-stage adaptation techniques on the three target datasets in term of CER%.

Target Dataset	A set of 150 adaptation samples		A set of 300 adaptation samples	
	One-stage	Two-stage	One-stage	Two-stage
MADCAT	32.02	**31.96**	30.94	**30.02**
AHTID/MW	13.53	**12.10**	10.40	**9.29**
IFN/ENIT (set a)	28.41	**27.01**	22.15	**20.33**
IFN/ENIT (set b)	29.23	**27.81**	22.73	**20.37**
IFN/ENIT (set c)	27.77	**26.37**	21.60	**19.61**
IFN/ENIT (set e)	36.47	**35.65**	33.90	**33.13**

Bold values represent the best achieved recognition performance

Table 5 Experimental results of the writer adaptation strategies carried on the MADCAT dataset for 15 writers (Test samples per writer : 150, total test samples : 2250)

Domain Adaptation strategy	Adaptation samples per writer	Total adaptation samples	CER%
Before Adaptation	NA[1]	NA	42.12
WIA	NA	300	30.02
WGA	300	4500	11.84
WGA and Augmentation	900	13500	**10.21**
WSA	300	4500	14.22
WSA and Augmentation	900	13500	12.95

[1] NA: Not Adapted

model is only trained on the source domain: KHATT dataset). The reason behind this low performance can be explained by the domain gap between the source and the target domains since text images are scanned using different devices and written by various writers with different civilizations. Moreover, text lines have different size in terms of width and height, as depicted from Fig. 7 (Sect 5).

The rates presented in Table 5 show that the WIA is an effective way to perform domain adaptation when we have not any knowledge of writers in the target dataset. However, we notice that the WGA leads to better results when compared to the WIA and WSA strategies. Using a total set of 300 samples per writer, the global adaptation (WGA) reduces the CER by 30.28 % compared to the obtained performance without adaptation. To explore the effect of the size of the adaptation set, we applied some data augmentation techniques to increase the amount of relevant adaptation data. In this context, we apply some affine transformations such as rotation, translation, scaling, and some morphological distortions such as erosion and dilation. Figure 8 shows some samples of the augmented data. The blue line indicates the distortion at the baseline level, while the orange box shows the scale variation introduced on the handwritten line images.

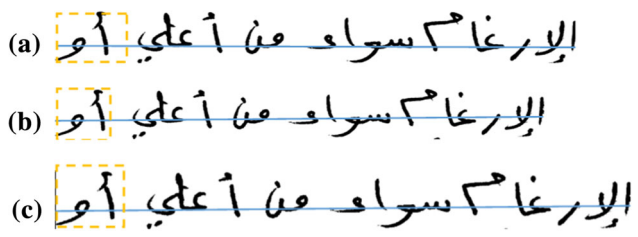

Fig. 8 Examples of augmented images extracted from the MADCAT dataset, **a** Original image, **b** rotated and down-rescaled image, c) rotated and up-rescaled image

We notice that using data augmentation in the WGA reduces the CER from 11.84 % to 10.21 %. The results of the WSA strategy have a similar behavior to the WGA since the WSA improves considerably the recognition results. Nonetheless, the obtained performance using the WSA is slightly lower compared to the WGA strategy. This result can be explained by the scarcity of some characters in the target dataset as the training is performed using writer specific data. However, using the WGA strategy, the data of all writers are merged to train the model, allowing to overcome the problem of data scarcity for some characters which consequently increases the recognition performance. For a better analysis of the performance of the different techniques, we show in Fig. 9 the obtained CER per writer. As it can be seen from this figure, the WGA with data augmentation technique performs better compared to the other proposed strategies for all writers.

6.3.2 Evaluation on the AHTID/MW dataset

To confirm the efficiency of the proposed writer adaptation strategies, we carried out additional experiments on the

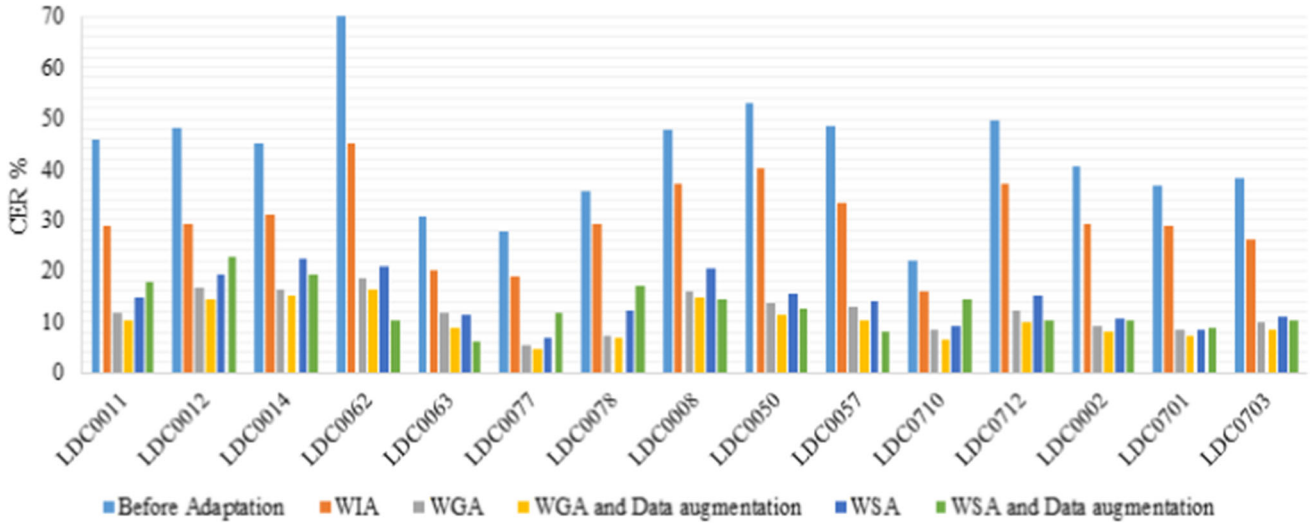

Fig. 9 Recognition results per writer of the different adaptation strategies evaluated on the MADCAT dataset

Table 6 Performance of the proposed adaptation strategies on the AHTID/MW dataset for 15 writers (Test samples per writer : 20, total test samples : 300)

Task	Adaptation samples per writer	Total adaptation for all writers	CER%
Before Adaptation	NA[1]	NA	22.08
WIA	NA	300	9.29
WGA	40	600	4.56
WGA and Augmentation	280	4200	**2.93**
WSA	40	600	14.21
WSA and Augmentation	280	4200	18.41

[1]NA: Not Adapted

AHTID/MW handwritten text lines dataset. The analysis of the results presented in Table 6 shows that the WIA adaptation enhances considerably the recognition results when the adaptation process does not include writers's information from the test set. The obtained results confirm also that the WGA is an effective way to perform the adaptation process compared to the use of WSA strategy. We notice that the WSA method integrating augmentation techniques leads to unsatisfactory results. This low performance can be explained by its sensitivity to the quantity of available data for adaptation. In fact, the accuracy of the WSA strategy strongly depends of the number of occurrences of each character in the adaptation set. When few character samples are available for a given writer, the optical model cannot learn efficient representation of the writer's style and cannot discriminate efficiently those characters. Consequently, we can confirm that the optimal method to minimize the gap between different source and target domains is the global adaptation since it allows to overcome the problem of data scarcity. We show in Fig. 10 the CER per writer using the different adaptation techniques.

6.3.3 Evaluation on the IFN/ENIT dataset

In this section, we present the obtained results on the IFN/ENIT dataset. It can be seen from the character error rates presented in Table 7 that the WGA performs better compared to the WSA, similarly to the previous experiments conducted on the MADCAT and AHTID/MW datasets. This result can be explained by the scarcity of writer's samples used in the adaptation process. With the integration of data augmentation techniques, the recognition results were slightly improved using the WGA strategy but not for the WSA method. In fact, data augmentation inserts some deformations in the text that change the writer's styles, and consequently the model cannot learn specific writer's representations that could enhance the recognition performance (Fig. 11). We show in Fig. 12 the achieved recognition performance per writer using the different adaptation techniques.

Fig. 10 Recognition results per writer of the different adaptation strategies evaluated on the AHTID/MW dataset

Table 7 Recognition results performed on the IFN/ENIT dataset (Test samples per writer: 20, total test samples: 300)

Task	Adaptation samples per writer	Total adaptation for all writers	CER%
Before Adaptation	NA[1]	NA	44.15
WIA	NA	300	22.99
WGA	64	960	7.97
WGA and Augmentation	280	6720	**7.75**
WSA	64	960	19.41
WSA and Augmentation	280	6720	23.74

[1] NA: Not Adapted

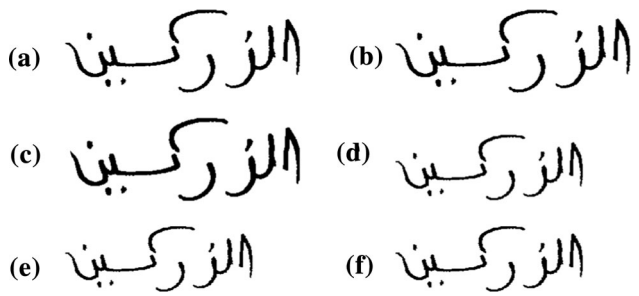

Fig. 11 Examples of augmented images extracted from the IFN/ENIT dataset, a) Original image, b–f) rotated and re-scaled image

6.4 Evaluation with McNemar's test

Lastly, in order to verify whether the pairwise differences in the performance of the different adaptation strategies are significant or not, we perform the McNemar's test [57]. This test is applied to determine whether a learning algorithm is more efficient than another. Once the difference is meaningful, it implies that the first method is clearly more efficient than the second. On the opposite, if the difference is not statistically relevant, then the difference in performance is insufficient to decide on the superiority of one method over another. So, firstly, we compute the contingency table [[N1,N2],[N3, N4]] in the assumption that there are two algorithms I and II where:

- N1 is the number of samples misclassified by both algorithms.
- N2 is the number of samples misclassified by algorithm I but not II.
- N3 is the number of samples misclassified by algorithm II but not I.
- N4 the number of samples correctly classified by both algorithms.

In Tables 8, 9 and 10, we present all the pairwise comparisons between every two strategies, and for each, we calculate the *p*-value, i.e. the probability that the null hypothesis is true for the three datasets. The smaller the p-value, the more the difference of performance is likely to be significant. We reject the null hypothesis in the case of the p-value is lower than 0.05. In this way, it is clear that the WGA method incorporating data augmentation techniques is significantly different from the other adaptation strategies performed on the three datasets, particularly, the WIA and WSA strategies.

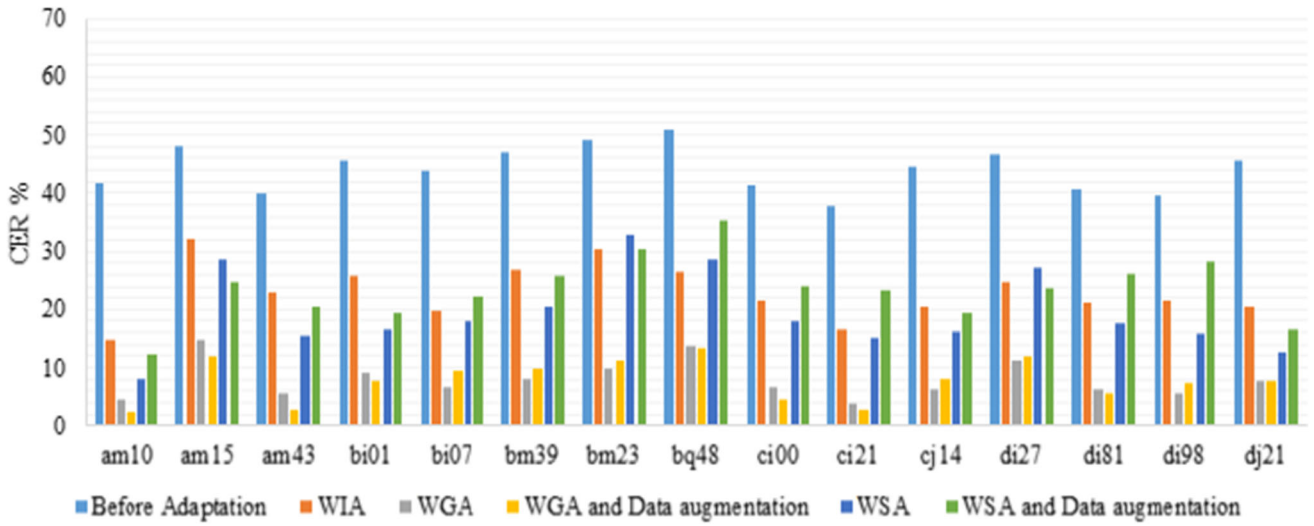

Fig. 12 Recognition results per writer of the different adaptation strategies evaluated on the IFN/ENIT dataset

Table 8 The *p*-values of McNemar's test for all the pairwise comparisons on the MADCAT dataset. NA: Not Adapted

Strategies	S1	S2	S3	S4	S5
S1: WIA	NA	1.07×10^{-9}	4.1×10^{-48}	3.94×10^{-31}	1.47×10^{-23}
S2: WGA		NA		4.85×10^{-36}	2.1×10^{-48}
S3: WGA + augmentation			NA	1.11×10^{-17}	2.88×10^{-10}
S4: WSA				NA	6.36×10^{-3}
S5: WSA + augmentation					NA

Table 9 The *p*-values of McNemar's test for all the pairwise comparisons on the AHTID/MW dataset. NA: Not Adapted

Strategies	S1	S2	S3	S4	S5
S1: WIA	NA	2.77×10^{-15}	4.47×10^{-35}	7.92×10^{-9}	5.82×10^{-11}
S2: WGA		NA	3.42×10^{-15}	1.58×10^{-30}	9.86×10^{-32}
S3: WGA + augmentation			NA	4.28×10^{-50}	2.67×10^{-51}
S4: WSA				NA	0.125
S5: WSA + augmentation					NA

Table 10 The p-values of McNemar's test for all the pairwise comparisons on the IFN/ENIT dataset. NA: Not Adapted

Strategies	S1	S2	S3	S4	S5
S1: WIA	NA	2×10^{-33}	7.8×10^{-36}	1.8×10^{-8}	6.55×10^{-5}
S2: WGA		NA	0.6	3.4×10^{-17}	3.7810^{-26}
S3: WGA + augmentation			NA	4.39×10^{-20}	8.41×10^{-28}
S4: WSA				NA	4.5×10^{-3}
S5: WSA + augmentation					NA

6.5 Comparison with state-of-the-art approaches

To perform a straightforward comparison with most of the existing systems in the literature, we have included a lexicon during the decoding process because most of the existing systems incorporate linguistic resources based generally on lexicons and/or language models. The decoding is performed using the Weighted Finite State Transducers (WFST) [58] allowing the integration of a word lexicon.

We compared our system with previous methods relying on Dynamic Bayesian Network (DBN), HMMs and Deep Neural Networks (DNN) models. It is worth noting that state-of-the-art systems evaluated on the IFN/ENIT dataset commonly use the a, b and c partitions (about 18K

samples) for training and the d partition for testing. However, we only use 1K samples increased to 8K through augmentation techniques and we keep the d-partition for testing. As it is shown in Table 11, our proposed adaptation strategy shows interesting performance compared to the other systems, although we use a reduced training set.

6.6 Discussion

The objective of this paper is to have a comprehensive and systematic study of different supervised domain adaptation strategies for the recognition of Arabic handwritten text images. In the literature, we noticed that there is no study that was focused on this problem when dealing with different source and target domains using deep learning models. From this study, we conclude that the WSA strategy is not generally suitable to enhance the recognition results since it strongly depends of the quantity of writer's data, which is generally hard to collect. One alternative to surpass this problem is the use of the WGA strategy. It was interesting also to draw attention to the influence of data augmentation techniques on the proposed writer adaptation approach since it improves the recognition results of the WGA strategy. However, we have observed that the incorporation of augmentation techniques is not efficient in the case of the WSA strategy. Despite the independent writer adaptation strategy achieves relatively lower performance compared to the writer global adaptation, we argue that it should be considered as an alternative as it is generally harder to collect a writer specific data.

Table 11 Performance comparison with state-of-the-art systems on the IFN/ENIT dataset

Model	WER %
HMM [59]	16.44
HMM [60]	9.04
HMM [61]	7.14
Adaptation HMM [62]	5.82
Multi-stream HMM [63]	20.4
Multiple classifier [64]	23.46
HMMs Fusion [65]	2.3
DBN [66]	21.5
Embedded HMM [67]	12.07
CNN-HMM [68]	11.05
SVM (statistical/geometric features) [69]	19.22
PHOG-SVM [70]	3.6
Multi-stage HMM [71]	2.29
Two-stream DNN [72]	5.06
Ours (WGA and Augmentation)	**6.13**

7 Conclusion

This paper proposed an extensive evaluation of domain and writer adaptation strategies for Arabic handwriting recognition task using a deep learning approach in a supervised way. To overcome the domains shift issue and writer's styles variation, we proposed different adaptation strategies to fine-tune a CNN-BLSTM-CTC architecture, especially, when few labelled data are available in the target domain. The adaptation is based on a two-stage transfer learning technique to extract the most relevant features modelling the handwritten text styles variety of the target dataset.

Three main adaptation strategies were proposed including independent-writer adaptation, global-writer adaptation and specific-writer adaptation. This paper has clearly shown that taking into account the writer's information during the supervised adaptation process improves considerably the recognition results. We have found that the specific writer adaptation strategy slightly improves the recognition results, but such a strategy is highly dependent of the quantity of data samples. An alternative to surpass this problem is the use of the global adaptation strategy, which is less sensitive to the number of samples per writer. The experimental study shows that the data augmentation technique improves the global adaptation performance, while it is inefficient in the case of specific adaptation. Our work is evaluated using three datasets which proves its effectiveness independently of the input level (line or word images). In our future research, we can explore the self-supervised training [73] to perform the global writer adaptation without the requirement of labelled data. Another perspective consists of exploring other recognizer architectures, specially Transformer based models.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Kessentini Y, Paquet T, Hamadou A B (2007) A multi-stream approach to off-line handwritten word recognition, in: international Conference on Document Analysis and Recognition, ICDAR, Vol. 7, p. 317-321
2. Bernard AB, Menasri F, El-Hajj R, Mokbel C, Kermorvant C, Likforman L (2011) Dynamic and contextual information in

HMM Behaviour for handwritten word recognition. IEEE Trans Pattern Anal Mach Intell 99:2066–2080
3. Koerich AL, Sabourin R, Suen CY (2003) Lexicon-driven HMM decoding for large vocabulary handwriting recognition with multiple character models. Int J Doc Anal Recognit, IJDAR 6:126–144
4. Graves A, Eck D, Beringer N, Schmidhuber J, Biologically plausible speech recognition with LSTM neural nets, in: Biologically Inspired Approaches to Advanced Information Technology, 2019, p. 127-136
5. Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, Schmidhuber J (2009) A novel connectionist system for unconstrained handwriting recognition. IEEE Trans Pattern Anal Mach Intell 31:855–68
6. Wollmer M, Metallinou A, Eyben F, Schuller B, Narayanan S S, (2010) Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional LSTM modeling, in: Biologically plausible speech recognition with LSTM neural nets, p. 2362-2365
7. Gers F A, Eck D, Schmidhuber J, (2001) Applying LSTM to Time Series Predictable Through Time-Window Approaches. Artificial Neural Networks, in: Artificial Neural Networks, ICANN, p. 669-676
8. Jaeger H (2002) Tutorial on Training Recurrent Neural Networks, Covering BPTT, RTRL, EKF and the Echo State Network' approach. Sankt Augustin. Tech Rep, Germany
9. Graves A, (2008) Supervised sequence labelling with recurrent neural networks. Ph.D. dissertation, Ph.D. dissertation, Technical University Munich,
10. Graves A, S Fernández, Gomez F, Schmidhuber J, (2006) Connectionist Temporal Classification : labelling unsegmented sequence data with recurrent neural networks, in: International conference on Machine learning, ICML, p. 369-376
11. Jemni S K, Kessentini Y, Kanoun S, Ogier J, (2018) Offline Arabic Handwriting Recognition Using BLSTMs Combination, in: IAPR International Workshop on Document Analysis Systems, DAS, p. 31-36
12. Cherawala Y, Roy PP, Cheriet M (2017) Combination of context-dependent bidirectional long short-term memory classifiers for robust offline handwriting recognition. Pattern Recognit Lett 90:58–64
13. Oprean C, Likforman-Sulem L, Popescu A, Mokbel C, (2015) BLSTM-based handwritten text recognition using Web resources, in: International Conference on Document Analysis and Recognition, ICDAR, p. 466-470
14. Jemni SK, Kessentini Y, Kanoun S (2020) Improving recurrent neural networks for offline arabic handwriting recognition by combining different language models. Int J Pattern Recognit Artif Intell. https://doi.org/10.1142/S0218001420520072
15. Jemni SK, Kessentini Y, Kanoun S (2019) Out of vocabulary word detection and recovery in Arabic handwritten text recognition. Pattern Recognit 93:507–520
16. Graves A, Schmidhuber J (2008) Offline handwriting recognition with multidimensional recurrent neural networks. Adv Neural Inform Process Syst 21:545–552
17. Louradour J, Kermorvant C, (2013) Curriculum Learning for Handwritten Text Line Recognition, arxiv preprint arxiv :1312.1737 1-9
18. Chherawala Y, Roy PP, Cheriet M (2016) Feature set evaluation for offline handwriting recognition systems: application to the recurrent neural network model. IEEE Trans Cybern 46:2825–2836
19. Castro D, Bezerra B L D, Valenca M, (2018) Boosting the deep multidimensional long-short-term memory network for handwritten recognition systems, in: International Conference on Frontiers in Handwriting Recognition, ICFHR, p. 127-132
20. Simard P Y, Steinkraus D, Platt J C, (2003) Best practices for convolutional neural net- works applied to visual document analysis, in: International Conference on Document Analysis and Recognition, ICDAR, p. 958-962
21. Eltay M, Zidouri A, Ahmad I (2020) Exploring deep learning approaches to recognize handwritten Arabic texts. IEEE Access 8:89882–89898
22. Voigtlaender P, Doetsch P, Ney H, (2016) Handwriting Recognition with Large Multidimensional Long Short-Term Memory Recurrent Neural Networks, in: International Conference on Frontiers in Handwriting Recognition, ICFHR, p. 228-233
23. Altwaijry N, Al-Turaiki I (2021) Arabic handwriting recognition system using convolutional neural network neural computing and applications. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05070-8
24. Bahdanau D, Cho K, Bengio Y, (2014) Neural machine translation by jointly learning to align and translate, in: arXiv:1409.0473, p
25. Li J, Jin K, Zhou D, Kubota L, Ju Z (2020) Attention mechanism-based CNN for facial expression recognition. Neurocomputing 411:340–350
26. Bluche T, Louradour J, Messina R (2017) Scan, Attend and Read: End-to-End Handwritten Paragraph Recognition with MDLSTM Attention, in: International Conference on Document Analysis and Recognition, ICDAR, p. 1050-1055
27. Michael J, Labahn R, Gruning T, Zollner J, (2019) Evaluating Sequence - to - Sequence Models for Handwritten Text Recognition, in: International Conference on Document Analysis and Recognition, ICDAR, p. 1286-1293
28. Le A D, Nguyen H T, Nakagawa M, (2020) End to End Recognition System for Recognizing Offline Unconstrained Vietnamese Handwriting, in: SN Computer Science, Vol. 7, pp. 1–8
29. T. Bluche, R. Messina, (2017) Gated Convolutional Recurrent Neural Networks for Multilingual Handwriting Recognition, in: Proceeding of International Conference on Document Analysis and Recognition (ICDAR), IEEE, pp. 646–651
30. Poulos J, Valle R (2021) Character-based handwritten text transcription with attention networks. Neural Comput Appl. https://doi.org/10.1007/s00521-021-05813-1
31. Shimodaira H (2000) Improving predictive inference under covariate shift by weighting the log-likelihood function. J Statist Plan Inference 90(2):227–244. https://doi.org/10.1142/S0218001420520072
32. Ponce J, Berg T L, Everingham M, Forsyth D A, Hebert M, Lazebnik S, Marszalek M, Schmid C, Russell B C, Torralba A, et al. (2006) J, Dataset issues in object recognition, in: Toward category-level object recognition, p. 29-48
33. Long M, Cao Y, Wang J, Jordan M I, (2015) Learning transferable features with deep adaptation networks, in: arXiv preprint arXiv:1502.02791 p
34. Motiian S, Jones Q, Iranmaesh SM, Doretto G, (2017) Few-Shot Adversarial Domain Adaptation, in: Conference on Neural Information Processing Systems, NIPS, p
35. Fernando B, Habrard A, Sebban M, Tuytelaars T, (2013)Unsupervised visual domain adaptation using subspace alignment, in: IEEE ICCV, p. 2960-2967
36. Fernando B, Habrard A, Sebban M, Tuytelaars T, (2016) Learning the roots of visual domain shift, in: Computer Vision-ECCV 2016 Workshops, p. 475-482
37. Zhang XY, Liu CL (2013) Writer adaptation with style transfer mapping. IEEE Trans Pattern Anal Mach Intel 35(7):1773–1787
38. Ahmad I, Fink G A, (2015) Training an arabic handwriting recognizer without a handwritten training dataset, in: nternational Conference on Document Analysis and Recognition, ICDAR, p. 476-480

39. Serrano JAR, Perronnin F, Sanchez G, Llados J (2010) Unsupervised writer adaptation of whole-word HMMs with application to word-spotting. Pattern Recogn Lett 31(8):742–748
40. Nosary A, Heutte L, Paquet T (2004) Unsupervised writer adaptation applied to handwritten text recognition. Pattern Recognit 37(2):385–388
41. Aradillas J C, Murillo-Fuentes J J, Olmos P M, (2018)Boosting Handwriting Text Recognition in Small Databases with Transfer Learning, in: International Conference on Frontiers in Handwriting Recognition, ICFHR, no. 429-434, https://doi.org/10.1109/ICFHR-2018.2018.00081
42. Lei K, Marçal R, Alicia F, Pau R, Mauricio V, (2020) Unsupervised adaptation for syntheticto-real handwritten word recognition, in: WACV,
43. Souibgui M A, Fornés A, Kessentini Y, Tudor C, (2021) A Few-shot Learning Approach for Historical Ciphered Manuscript Recognition, in: International Conference on Pattern Recognition, ICPR, pp. 5413–5420
44. Granet A, Morin E, Mouchere H, Quiniou S, Gaudin C V, (2018) Transfer learning for handwriting recognition on historical documents, in: International Conference on Pattern Recognition Applications and Methods, ICPRAM, p. 432-439
45. Yang H M, Zhang X Y, Yin F, Sun J, Liu C L, (2018) Deep transfer mapping for unsupervised writer adaptation, in: International Conference on Frontiers in Handwriting Recognition, ICFHR, p. 151-156
46. Kang L, Rusinol M, Fornés A, Riba P, Villegas M, (2020) Unsupervised Adaptation for Synthetic-to-Real Handwritten Word Recognition, in: IEEE Winter Conference on Applications of Computer Vision, WACV, p. 3491-3500
47. Puigcerver J, (2017) Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition?, in: International Conference on Document Analysis and Recognition, ICDAR, p. 67-72
48. Miseikis J, Brijačak I, Yahyanejad S, Glette K, Elle O J, Torresen J, (2019) Two-Stage Transfer Learning for Heterogeneous Robot Detection and 3D Joint Position Estimation in a 2D Camera Image Using CNN, in: International Conference on Robotics and Automation, ICRA, p
49. Oquab M, Bottou L, Laptev I, Sivic J, (2014) Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks, in: IEEE Conference on Computer Vision and Pattern Recognition, p. 1717-1724
50. Graves A, Fernandez S, Gomez F, Schmidhuber J, (2006) Connectionist temporal classifcation: Labelling unsegmented sequence data with recurrent neural' networks, in: ICM of the International Conference on Machine Learning, p. 369-376
51. Mahmoud S A, Ahmad I, Alshayeb M, Al-Khatib W G, Parvez M T, Fink G A, Margner V, El Abed H, (2012) KHATT: Arabic offline handwritten text database, in: International Conference on Frontiers in Handwriting Recognition, ICFHR, no. 449-454,
52. Strassel S, (2009) Linguistic resources for Arabic handwriting recognition, in: International Conference on Arabic Language Resources and Tools, no. 37-41
53. Mezghani A, Kanoun S, Khemakhem M, (2012) A Database for Arabic Handwritten Text Image Recognition and Writer Identification, in: International Conference on Frontiers in Handwriting Recognition, ICFHR, no. 399-402
54. Pechwitz M, Maddouri S S, Margner V, Ellouze N, Amiri H, (2002) IFN/ENIT-database of handwritten Arabic words, in: Colloque International Francophone sur l'Ecrit et le Document, CIFED, no. 129-136,
55. V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, in: Soviet physics doklady, Vol. 10, Soviet Union, 1966, pp. 707–710
56. Kingma D P, Ba J, (2015) Adam: A Method for Stochastic Optimization, in: International Conference for Learning Representations, p
57. Dietterich TG (1998) Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput 10(7):1895–1923. https://doi.org/10.1162/089976698300017197
58. Mohri M, Pereira F, Riley M (2002) Weighted finite-state transducers in speech recognition. Comput Speech Lang 16(1):69–88. https://doi.org/10.1006/csla.2001.0184
59. M. Pechwitz, V. Maergner, (2003) Hmm based approach for handwritten arabic word recognition using the ifn/enit - database, in: International Conference on Document Analysis and Recognition, pp. 890–894. https://doi.org/10.1109/ICDAR.2003.1227788
60. Al-Hajj R, Mokbel C, Likforman-Sulem L, (2007) Combination of HMM-based classifers for the recognition of Arabic handwritten words, in: Proceeding of International Conference on Document Analysis and Recognition (ICDAR), pp. 959–963
61. P. Dreuw, S. Jonas, H. Ney, (2008) White-space models for offline arabic handwriting recognition, in: 2008 19th International Conference on Pattern Recognition, pp. 1–4. https://doi.org/10.1109/ICPR.2008.4761841
62. P. Dreuw, D. Rybach, C. Gollan, H. Ney, (20090) Writer adaptive training and writing variant model refinement for offline arabic handwriting recognition, IEEE Computer Society, USA. https://doi.org/10.1109/ICDAR.2009.9
63. Kessentini Y, Paquet T, Ben Hamadou A (2010) Off-line handwritten word recognition using multistream hidden Markov models. Pattern Recognit Lett 31:60–70
64. AlKhateeb JH, Ren J, Jiang J, Al-Muhtaseb H (2011) Offline handwritten arabic cursive text recognition using hidden markov models and re-ranking. Pattern Recognit Lett 32(8):1081–1088. https://doi.org/10.1016/j.patrec.2011.02.006
65. Azeem SA, Ahmed H (2013) Effective technique for the recognition of offline arabic handwritten words using hidden markov models. Int J Doc Anal Recognit, IJDAR 16(8):399–412. https://doi.org/10.1109/ACCESS.2020.2994248
66. Jayech K, Mahjoub M, Amara N (2016) Arabic handwritten word recognition based on dynamic bayesian network. Int Arab J Inf Technol 13:1024–1031
67. Rabi M, Amrouch M, Mahani Z (2018) Recognition of cursive arabic handwritten text using embedded training based on hidden markov models. Int J Pattern Recognit Artif Intell 32(01):1860007. https://doi.org/10.1142/S0218001418600078
68. Amrouch M., Rabi M., Es-Saady Y., (2018) Convolutional Feature Learning and CNN Based HMM for Arabic Handwriting Recognition, in: Image and Signal Processing, ICISP, Lecture Notes in Computer Science, Vol. 10884, pp. 5413–5420. https://doi.org/10.1007/978-3-319-94211-7_29
69. Tavoli R, Keyvanpour M, Mozaffari S (2018) Statistical geometric components of straight lines (sgcsl) feature extraction method for offline arabic/persian handwritten words recognition. IET Image Process 12(9):1606–1616
70. T. M. Ghanim, M. Khalil, H. M. Abbas, (2019) Multi-stage offline arabic handwriting recognition approach using advanced cascading technique, in: ICPRAM,
71. Ahmad I, Fink GA (2019) Handwritten arabic text recognition using multi-stage sub-core-shape hmms. Int J Doc Anal Recognit 22:329–349. https://doi.org/10.1007/s10032-019-00339-8
72. Sulaiman A, Omar K, Nasrudin MF (2021) Two streams deep neural network for handwriting word recognition. Multim Tools Appl 80(8):5473–5494. https://doi.org/10.1007/s11042-020-09923-1

73. Bhunia AK, Chowdhury PN, Yang Y, Hospedales T, Xiang T, Song YZ (2021) Vectorization and rasterization: Self-supervised learning for sketch and handwriting, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)