

# A new gated multi-scale convolutional neural network architecture for recognition of Persian handwritten texts

Sara Khosravi, Abdoloh Chalechale\*

*Department of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran*

*(Communicated by Ehsan Kozegar)*

---

## Abstract

Due to the ease of writing by hand and the inherent interest in it, writing by hand is still popular among many people. Considering the digitization of today's world and the massive amount of current information on paper, there is a need for a system to convert handwriting into its digital form to speed up access to information and reduce storage space. According to the research carried out in this field, recognizing Persian handwritten texts remains a relatively difficult issue due to the complex and irregular nature of writing and the diversity of people's handwriting. This research introduces a novel method to recognize handwritten texts at the sentence level. To use word recognition methods in sentence recognition, segmentation techniques are needed to separate the words in the sentence. The segmentation algorithm in handwritten texts is inefficient due to overlapping words. Since Recurrent Neural Networks (RNN) were a turning point in the recognition of correct writing, in this article, by removing the segmentation step, a new architecture, an RNN combined with a Gated Multi-scale Convolutional Neural Network (GMCNN), is introduced in order to recognize handwritten sentences. Using the proposed architecture, recognizing Persian handwritten sentences in the Sadri dataset has a character error rate of 2.99%, a word error rate of 6.67%, and a sentence error rate of 36.87%. For further evaluation, the proposed method was also evaluated on IAM and Washington datasets. The results show that the proposed method outperforms other known algorithms.

Keywords: handwritten text recognition, convolutional neural network, recurrent neural network, connectionist temporal classifier  
2020 MSC: 68T07

---

## 1 Introduction

Writing is a complex process of reflecting the perception and communication of the outside world in the human mind, which can inspire new thoughts and imaginations. In fact, with the movement of the hand in writing, a connection is established between the sense of touch and the human brain, which stimulates thinking and increases learning. Organizing thoughts on paper reduces mental worries and makes people concentrate and relax. The limitations of writing on paper are fewer than digital writing. Due to the interest and ease of writing by hand, many people still prefer to handwrite texts instead of digital writing. Another noteworthy point is that the manuscript can express a society's identity, feelings, and culture.

---

\*Corresponding author

Email addresses: [khosravi\\_un@pnu.ac.ir](mailto:khosravi_un@pnu.ac.ir) (Sara Khosravi), [chalechale@razi.ac.ir](mailto:chalechale@razi.ac.ir) (Abdoloh Chalechale)

On the other hand, digital and texts can be processed, making accessing and maintaining these resources easier. Human's innate desire to write and the existence of a considerable amount of information on paper prompted researchers to provide methods to implement a system for recognizing handwritten texts and converting them into digital texts. The process of converting printed scanned images or handwritten texts into a format to be processed by the computer is called Optical Character Recognition (OCR); So that they can be electronically edited, searched, stored, and made more compact and efficient [19].

OCR can be divided based on the type of input data (online and offline) and recognition text (printed and handwritten). According to the conditions and type of usage, the data required for training and testing each of these recognition systems will be different. These collections can start from small written units such as numbers and individual letters and extend to Handwritten Text Recognition (HTR) [1, 16, 22, 30].

In the age of digitization, HTR plays an essential role in information processing. Processing digital files is faster and cheaper than processing traditional paper files. On the one hand, the desire for handwriting and, on the other, the need to convert them into digital texts in various fields such as doctors' prescriptions, judges' verdicts, professors' pamphlets, texts and converting notes written with light pens into digital formats to organize and edit them, have shown their uses and which resulted in the Handwritten text recognition to become one of the popular research fields. Therefore, an efficient recognition system with large datasets, high recognition accuracy, and less computational complexity is required to convert handwritten texts into machine-readable formats [5, 17, 19, 24, 29].

Using deep neural networks, the recognition process at different levels of text segmentation, such as character, digits, word, line, and even paragraph levels, has evolved significantly. However, in the field of HTR, more research is needed to reach more satisfactory results. One solution in this field is to perform text decoding and post-processing using Natural Language Processing (NLP) techniques.

This manuscript was designed and implemented with the aim of handwritten sentence recognition in different languages, especially Persian. The proposed architecture by this method includes the use of Gated Multi-scale Convolutional Neural Network (GMCNN) blocks along with Recurrent Neural Network (RNN) blocks. The gate mechanism is also used in the convolutional layers and recent deep-learning approaches in the proposed model. According to the research, using the gate mechanism in the convolutional layers significantly increases the power of feature extraction and the final recognition accuracy. In the proposed optical model, the connectionist time classifier is used, and the goal is to minimize the validation losses. For this purpose, we dynamically used two optimizers, the first at the beginning of the training process, which has a more accurate and faster convergence capability, and the second at the end of the training process, which has a better discovery capability. The contribution of the article is described below.

1. Designing an innovative architecture of a gated multi-scale convolutional neural network to recognize handwritten sentences for integrated recognition of handwritten sentences by removing the segmentation step.
2. Presenting a unique technique to identify Persian/Arabic sentences using a convolutional network.
3. For the first time, the standardization of the Persian dataset includes removing or correcting inappropriate images, preparing labels and dividing it into three categories of training, validation, and testing, and using it to identify Persian sentences with the proposed method.

Coming up this article is organized as follows: Section 2 deals with the background of the research. In Section 3, the proposed method is comprehensively explained. In Section 4, the datasets used are described and explained in more details. Section 5 discusses the results of recognition and comparisons, and finally, Section 6 presents the conclusion and future work.

## 2 Related work

Usually, the process of automatic text recognition includes four steps [2]: (1) document digitization to obtain an image of each page of the document in electronic format; (2) dividing each page into areas corresponding to lines of text.; (3) identifying each line of text, and finally, (4) using a dictionary or language model to correct mistakes in text recognition and also in combining entire texts from lines obtained from step (3).

Due to the novelty of text-based recognition methods, few researches have been done in reviewing these methods. Reference [7] is one of the complete research works that has limited its scope to review the work done in the Scene Text Recognition (STR) field. The general framework in text-based recognition methods is to recognize the line or word and focus on mapping the entire text image to a sequence of letters and numbers. This work is done directly using an Encoder-Decoder Network; therefore, there is no need to separate the letters.

Optical models can be used with language models to minimize the challenges created in text recognition to help decode the text. Therefore, with the aim of results improvement, a dictionary of characters and words is generated from the dataset, and language limitations are created in the recognition process. To solve this problem, the researchers in [23] propose spelling correction techniques for text post-processing to achieve better results and remove the linguistic dependency between the optical model and the decoding stage.

The work done in [10] includes a hybrid architecture of CNN and RNN. This work aims to automatically recognize the text in the documentary images obtained by mobile phones. In the proposed method, pre-processing, feature extraction, and classification steps are integrated to recognize the images' text. The pre-processing step is applied to locate the text region and then divides that region into text line images. A sliding window divides the text line image into a sequence of frames in the second step. A CNN model is then used to extract features from each frame. Finally, an architecture that is a combination of bidirectional RNN, Gated Recurrent Units (GRU) block, and Connectionist Temporal Classification (CTC) [23] layer is considered to ensure the classification stage. Experimental results show that the proposed method can achieve promising recognition rates.

Off-line handwritten text recognition from images is an important problem for companies trying to digitize large volumes of scanned handwritten documents and reports. In [8], a new approach is introduced, which combines a deep convolutional network as a feature extractor with a recurrent encoder-decoder network to map an image to a sequence of characters corresponding to the text in the image. The used RNN-based encoder-decoder network basically performs the task of generating a target sequence from a source sequence. To increase the decoding capacity of the model, the beam search algorithm has been used, which searches for the best sequence from a set of hypotheses based on the joint distribution of individual characters. The introduced model takes a sampled version of the original image as input and thus makes it computationally and memory efficient. The obtained results showed that the encoder-decoder network had a significant increase in accuracy compared to the standard RNN-CTC formula.

It should be noted that the encoder-decoder model, which is also known as the Sequence-to-Sequence (Seq2Seq) model is an approach to neural networks which consists of two RNNs. The first RNN is responsible for encoding the input of variable-length symbol sequences into a fixed-length representation, While the second RNN decodes fixed-length symbols with other variable-length symbol sequences [23].

A potential problem with the Seq2Seq approach is that a neural network must be able to compress all the information needed from an input sentence to be decoded later. This can make it difficult for the model to process very long sentences, especially those that are longer than training sentences. According to [3], the attention mechanism allows the model to learn how to produce a content vector, which shows the input sentence in the best way at each stage of decoding. This means that the model learns to focus and pay more attention to relevant parts of the input sequence.

In [12], a hybrid handwritten text recognition model using deep neural networks that use the Seq2Seq approach has been suggested. This model uses a combination of significant features of CNN and RNN networks with LSTM. In the proposed method, the features extracted by CNN are later modeled with the Seq2Seq approach and given to the RNN-LSTM network to encode the visual features and decode the sequence of letters in the handwritten image. The proposed model has been tested with IAM and RIMES handwritten datasets and has performed better in letter and word accuracy than other methods.

Also, in [28], using the scientific branch of NLP, a three-step algorithm for Persian text recognition based on recognizing Persian sentences is presented. In the first step, this algorithm uses one of the methods of tagging connected components in order to separate sub-words. In the second step, the extracted sub-words are put together; And all meaningful words and then potential meaningful sentences are formed. In the last step, gram-2 and gram-3 language models, as well as Persian grammar rules, are used in order to recognize the correct sentence from a set of sentences. The results showed that by applying the proposed method, most of the Persian sentences could be recognized.

Another solution to solve the challenges of HTR is to provide a new method called Keyword Spotting (KWs), which is much more practical than an OCR-based solution. In [4], a new KWS method without learning is proposed, which can be applied to a heterogeneous set of handwritten documents. This work includes the introduction of a new profile-matching method to compare query word profiles (both top and bottom) with target word profiles.

Chinese handwritten text recognition has always been one of the most challenging tasks due to the existence of issues such as a large number of characters, complex structure, and different sizes, as well as issues caused by text images in different scenes. In order to address these challenges in [31], an end-to-end recognition method based on Convolutional Recurrent Neural Networks (CRNNs) is proposed to recognize texts in the scene, including Chinese texts. The proposed algorithm adopts a network framework that combines CNN, RNN, and CTC. CNNs are used to automatically extract feature sequences for each input image, which are from different convolutional layers. Then,

RNNs predict each frame of the feature sequence extracted from CNNs. The last part of the network is the transcription layer, which is responsible for converting predictions from RNN into true labels. In particular, the use of an asymmetric convolutional network, as well as feature reuse separately, can enhance the ability to extract features of horizontal text regions and detailed feature information in the image.

In [2], the problem of recognizing handwritten text in historical documents has been addressed using two deep learning techniques: Transfer Learning (TL) and Data Augmentation (DA), where few labeled examples are available, and some contain errors in the training set. In the proposed method, lines with wrong labels are identified and removed from the training set. In addition, mismatch-type errors are resolved by searching for real tags in the dataset. The main contributions in this work are 1- Analyzing how to perform TL, from a huge dataset to a smaller historical dataset, 2- Investigating which layers of the model need to be fine-tuned (which layers should be preserved) and which ones need to be retrained). 3- Analyzing methods for an effective combination of TL and DA, 4- Proposing an algorithm identify and reduce the effects of incorrect labeling in small training data sets and to fix wrong labels in some scenarios.

Sequential architectures are suitable for modeling lines of text, not only because of the inherent temporal aspect of the text but also to learn probability distributions over sequences of characters and words. However, using such recurrent patterns in the training phase is costly; Because their consecutive pipelines prevent parallelization. In [15], a new and non-repetitive method inspired by transformer models is introduced to solve the problem of handwritten text recognition; where it truncates any recurrent path during the training process (removes any duplicate network). Transformers rely entirely on attention mechanisms and ignore any repetitive pattern. The use of transformers in various language and vision applications shows a higher performance than recurrent networks; While with greater parallelism and thus reducing training times, they are superior to BLSTM or GRU.

### 3 The proposed method

The handwritten sentence recognition system takes the image as input and returns the entire recognized text. Figure 1 shows how the proposed recognition method works. The adopted datasets are built with a focus on the text recognition process; which are applied in the proposed method. For HTR, the partitioning method (training, validation and testing) follows what is defined by each set.

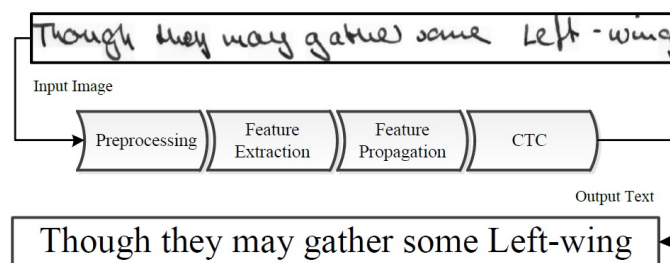


Figure 1: Overall scheme of the proposed sentence recognition method

In this paper, a new architecture including gated multi-scale convolutional neural network with RNN and CTC classifier will be introduced to recognize handwritten sentences in which, the recurrent multi-scale convolutional network is used to extract the features and produce the suitable probability matrix for CTC classification in the next stage; And in the last step, the Beam search algorithm will be used to convert the matrix to the textual output. In this method, the sentences are recognized entirely way without the need for segmentation.

#### 3.1 Preprocessing

Great variety in the writing style of people and external factors such as writing tools and environmental factors cause irregularities in the images. To normalize these irregularities, pre-processing techniques are used. For this purpose, we go through the following steps to minimize the diversities in the dataset images.

**Contrast correction [6]:** it is adopted to reduce illumination irregularities such as light peaks and excessive shadows. Therefore, techniques such as illumination compensation are different for this scenario; this technique is not very useful in black and white images.

**Slant correction [32]:** this is done with the aim of normalizing the vertical slope of the characters in the images. This method detects the tilt angle of the text on the vertical axis and then applies a geometric image transformation to adjust the deleted angle.

**Standardization [27]:** it is done with the purpose of changing the scale of pixels to create an image with pixels with mean and variance values of 0 and 1 respectively.

**Background estimation and size normalization:** the size of all input images must be  $128 \times 1024$  pixels. The issue regarding the image of words can be easily resolved with the process of resizing. But for the sentence images, the size of the sentences can vary from one to multiple words; the resizing operation causes a lot of image distortion. To solve this problem, we first estimate the background by averaging the pixels, and transfer the image of the sentence into another image with the estimated background of  $128 \times 1024$  dimensions.

**Tokenization [26]:** in the preprocessing stage, the text should be regularized as much as possible. Therefore, only standard tokenization is used to analyze the dataset. For this aim, the tokenization process for text data, which includes correcting the spacing, alignment, parenthesis, dashes and apostrophes, removing undefined characters and misplaces punctuation marks is done.

**Data augmentation:** with operations such as rotation, length and width shift, resizing new images are prepared for training. By learning new features from these images in addition to original images, the network is able to have an extended understanding of each object and become generalized so to speak.

### 3.2 The proposed architecture

The proposed optical model architecture in [34] is responsible for recognizing text from images and consists of gated multi-scale convolutional neural network blocks along with recurrent neural network blocks. In the proposed model, the Gated mechanism provided by [9] is also used in convolutional layers and recent deep learning approaches. It is well shown in the article [9]; that the use of the Gated mechanism in the convolutional layers significantly increases the feature extraction ability and the final recognition accuracy. GMCNN blocks are composed of the following layers:

- $3 \times 3$  Convolutional layer having 16 gated filters, concatenated with  $5 \times 5$  convolutional layer having 16 gated filters.
- $3 \times 3$  Convolutional layer having 32 gated filters, concatenated with  $5 \times 5$  convolutional layer having 32 gated filters.
- $2 \times 4$  Convolutional layer having 40 gated filters, concatenated with  $3 \times 5$  convolutional layer having 40 gated filters.
- $3 \times 3$  Convolutional layer having 48 gated filters, concatenated with  $5 \times 5$  convolutional layer having 48 gated filters.
- $2 \times 4$  Convolutional layer having 56 gated filters, concatenated with  $2 \times 4$  convolutional layer having 56 gated filters.
- $3 \times 3$  Convolutional layer having 64, concatenated with  $5 \times 5$  convolutional layer having 64.

For all convolutional layers, batch renormalization [14] is applied, and in the last three with gated mechanisms, dropout (rate 0.2) is applied. As activation and initialization, we use Heuniform and Parametric Rectified Linear Unit (PReLU) [13], respectively. Finally, the recurrent blocks consist of two BGRUs (128 hidden units per GRU) with dropout (rate 0.5) per GRU [9], alternating with a dense layer. Figure 2 shows the general view of the proposed architecture of the gated multi-scale convolutional neural network whit recurrent neural network.

The features extracted from convolutional blocks have a direct relationship with the size of their filters. If the size of the filters is chosen large; The details of the images cannot be seen in the convolutional operation of the filters; And vice versa, if the size of the filters is chosen small; The generalities of the images are ignored in the convolutional operations of the filters. Considering that in recognizing handwritten sentences, the input images contain very important details and generalities; And both of these are vital to identify the output word (in most languages and especially in Persian/Arabic languages, sometimes a very small dot changes the meaning of the word completely). Therefore, in each step of convolutional layers, both groups of filters should be used.

As shown in the article [33]; If the information of different convolutional layers is concatenated together instead of being multiplied; The final accuracy of the network improves. Therefore, in the architecture of the proposed model,

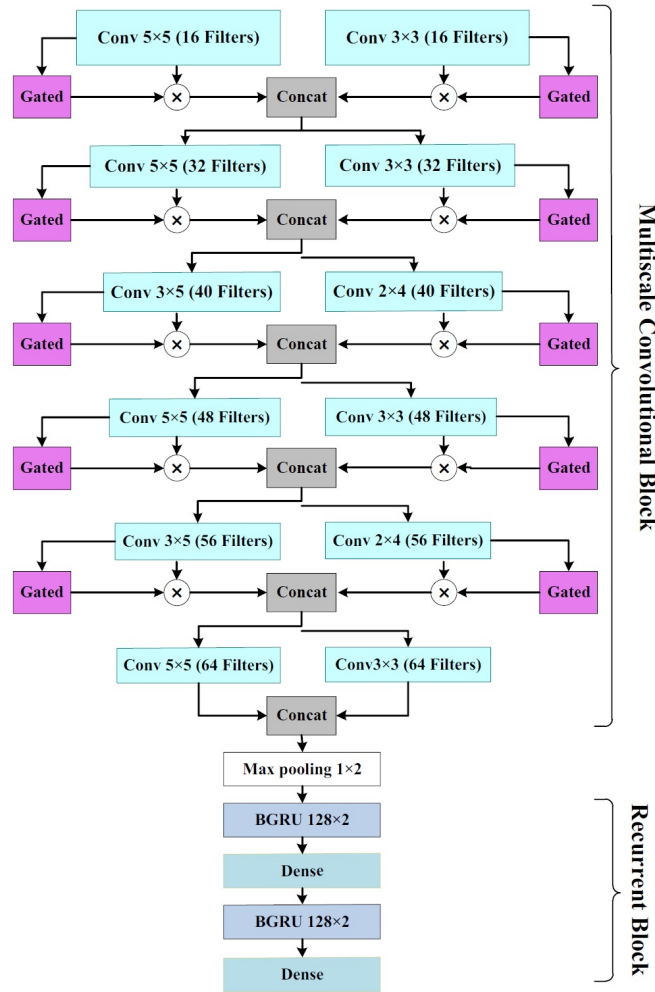


Figure 2: The proposed architecture of GMCNN with RNN

we concatenate the information of both groups of multi-scale filters. In Table 1, the output images from filters with different dimensions are compared with each other. As can be seen, by increasing the size of the filter, the details of the images are removed and only their general remains. For example, in the  $5 \times 5$  size filter, some points of the input image are removed; which means; that the features related to this part of the image cannot be seen in the output of the filter, and a smaller size filter should be selected to consider these parts.

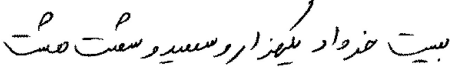
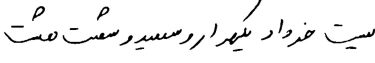
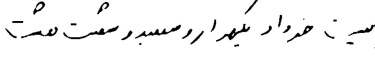
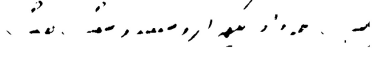
### 3.3 Combining the proposed architecture with CTC

The training and classification stages are performed for four optical models using Connectionist Temporal Classification. In this method, we train each model to minimize the value of CTC validation loss. For this purpose, we dynamically used two optimizers RMSprop [20] and Adam [18] with a learning rate of 0.0005 and mini-batches with the size of 16 image samples in each step. The way these two optimizers work is that for more accurate and faster convergence, at the beginning of the training process, the Adam optimizer, which has a better discovery capability, and at the end of the training, the RMSprop optimizer, which has a more suitable exploitation capability, are used.

Plateau learning rate reduction (coefficient 0.1) and early stopping mechanisms are also applied after 15 and 20 epochs, respectively, without improvement in the validation loss value. In addition, optical models have an input size of  $1 \times 128 \times 1024$  (height  $\times$  width  $\times$  channel) and a maximum text sentence length of 128 characters. Finally, the CTC function uses 97 printable characters from the ASCII code (including letters, numbers, punctuation marks) along with a null character (no decision) which is a total of 98 characters to decode the final text; This means that Accented Characters (not normalized) characters are converted to non-Accented (normalized) characters.

The classification using the CTC method is as follows; that the designed multiscale convolutional network moves its sliding window (filter) 128 times along the image; And at each location, it returns one probability for each of the

Table 1: Output images of filters with different dimensions

Median filter size	The output image from the median filter
Input image	
3 × 3	
5 × 5	
7 × 7	
9 × 9	

98 characters in the set. Finally, the CTC block converts this 128 × 98 matrix into a text sentence. Figure 3 shows the combination of the proposed architecture with connectionist temporal classifier; which includes the following blocks.

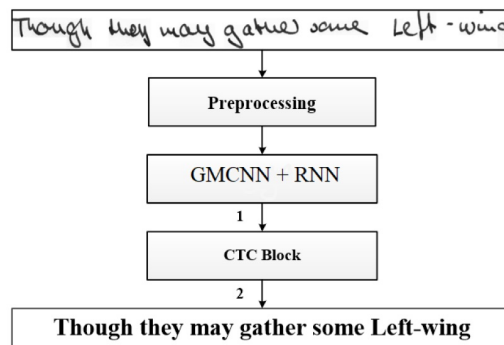


Figure 3: Combination of the proposed architecture with Connectionist Temporal Classification

Output of part 1: This part includes the output of the CNN+RNN network and as defined by the network, it is a 128 × 98 matrix; that each row shows the possibilities of all characters for the corresponding location. In Figure 4, the output matrix is shown along with the corresponding sliding window.

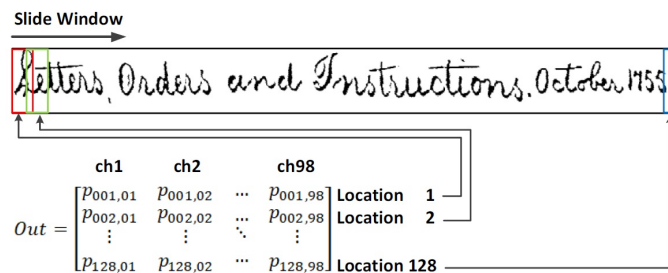


Figure 4: Output matrix of GMCNN + RNN block

Output of part 2: this part includes the decoding of the probability matrix; which is done by the CTC block. Various

algorithms have been provided for this purpose. The maximum probability method is an obsolete and unusable method. In this method, for each location, we calculate the most probable character according to the relevant probabilities and then convert the output to the final output with post-processing methods. In the proposed method, we used the Beam search algorithm to decode the probability matrix in the CTC block. In the Keras library of the Python programming language, a special function is provided for the Beam search method; By giving the probability matrix as input to the function, the output characters are calculated.

Since in convolutional networks, images are scanned from left to right to recognize sentences their corresponding tag (GT) should be filled with a blank character equal to the maximum length of the sentence. This issue causes errors when recognizing Persian/Arabic sentences due to the way of writing from right to left, and the mismatch of the image with its label. To solve this problem, a unique technique, inverting on the vertical axis, was used to recognize Persian/Arabic sentences. An inverted sample image is shown in Figure 5.

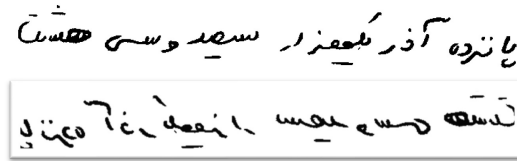


Figure 5: An inverted sample image along with vertical axis in Sadri.Dates dataset

## 4 Datasets

To evaluate the handwritten sentence recognition method, several public and available datasets including Sadri [25], IAM [21], and Washington [11] are used in Farsi and English. We will explain them below.

### 4.1 Sadri dataset

The Sadri dataset [25] includes numbers with separated handwritten digits, connected digits, dates, words, names, alphabets, free text, signs and mathematical symbols. The number of words in the Sadri dataset is 70,000 words, the number of figures is about 9,000 images, and the number of letters is based on different forms that they may have depending on their position in a word; There are 42962 images. This dataset also includes the number of 1999 handwritten sentences of solar dates. The data distribution is summarized in Table 2. A sample of handwritten sentences from this dataset is shown in Figure 6.

Table 2: Data distribution in Sadri dataset

Sentence Length		Average Tokens/Sentence		Partitioning			
Minimum	Maximum	Chars	Words	Training	Validation	Testing	Total
11	51	36	8	1399	299	301	1999

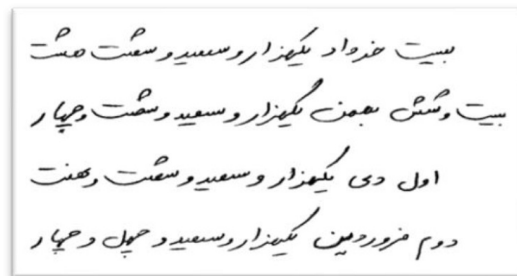


Figure 6: A sample of handwritten sentence images in Sadri dataset



### 4.2 IAM dataset

1539 pages of English handwritten text were scanned in gray scale and stored in the Department of Computer Science and Applied Mathematics (IAM) dataset [21]. It was discovered from these scanned pages that this collection was written by 657 different authors. Additionally, some of the images have darkness surrounding the words while others have a clear background. The partitioning for the suggested HTR method consists of 6161 lines for training, 900 lines for validation, and 1861 lines for testing. There are 78 and 9087 distinct words, respectively. The distribution of the data is shown in Table 3. Figure 7 displays illustrations of handwritten sentence examples from this dataset.

Table 3: Data distribution from IAM

Sentence Length		Average Tokens/Sentence		Partitioning			
Minimum	Maximum	Chars	Words	Training	Validation	Testing	Total
20	81	44	7	6161	900	1861	8922

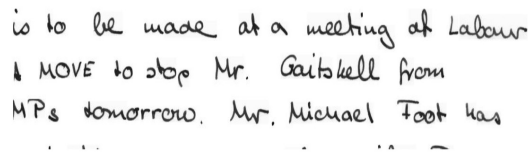


Figure 7: Examples of IAM dataset images

### 4.3 Washington dataset

The English-language articles written by George Washington in the 18th century that are stored in the Library of Congress made up the Washington dataset. This collection, which includes line-by-line transcription, is drawn from the historical manuscripts of two authors. It has fewer data (656 in total), which increases overfitting. Also, the images are also binary and normalized. Additionally, there are 1189 distinct words and 68 characters in this collection. The data is thus split into 325 lines for training, 168 lines for validation, and 163 lines for testing for HTR. The distribution of the data is shown in Table 4. Figure 8 displays illustrations of handwritten sentence examples from this dataset.

Table 4: Data distribution from Washington dataset

Sentence Length		Average Tokens/Sentence		Partitioning			
Minimum	Maximum	Chars	Words	Training	Validation	Testing	Total
4	62	42	7	325	168	163	656

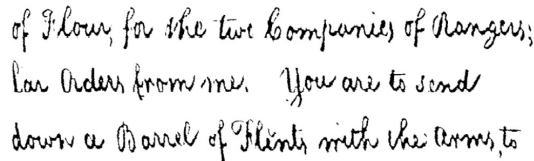


Figure 8: Examples of Washington dataset images

## 5 Results and Discussion

All codes of this paper are implemented on the Google Colaboratory platform in Python programming language. We will discuss the proposed algorithm’s performance regarding precision and execution time. Additionally, to demonstrate the superiority of the proposed approaches, each simulation’s results are compared with those of other approaches. To further show the mixed states of results, we organize the components of each experiment into groups. Figure 9 is a demonstration of how groups are organized as well as their components.

1. Datasets are responsible for the provision of images and texts.
2. The Optical model is responsible for rewriting the contents of images in the form of text.

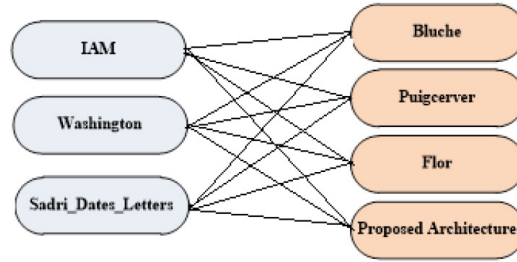


Figure 9: All datasets (first column) are tested with all the architectures (second column).

In this section, we present all the results of each dataset in the experiments. We use Character Error Ratio (CER), Word Error Ratio (WER), and Sentence Error Ratio (SER) [15] and compare the results of our work to those of other works.

### 5.1 Results of different architectures on datasets for recognition task

As demonstrated in Table 5, on the Washington, and Sadri Dates datasets, for the evaluation metrics of WER, CER, and SER, the Flor architecture showed better performance than Bluche and Puigcerver [23]. As derived from the Washington dataset shows a decrease of 1.8%, 0.75%, and 9.35% in CER, WER, and SER. This decreasing trend is repeated for Sadri Dates, with values of 0.67% in CER, 1.39% in WER, and 7.98% in SER. Moreover, on the IAM dataset, Puigcerver surpassed Bluche and Flor’s architectures; meanwhile, compared to mentioned architectures, our proposed architecture could reduce the CER, WER, and SER by about 1.77%, 4.66%, and 18.92%, respectively. Therefore, with due regard to obtained results, the precision of the proposed architecture on most of the datasets has witnessed a significant improvement in handwritten sentence recognition. Table 8 presents some outputs of recognition performed on different datasets with their labels.

Table 5: Recognition outputs of different architectures on datasets.

DataSet	Error Rate	Flor	Puigcerver	Bluche	Proposed
IAM	CER (%)	5.8	5.8	9.9	4.08
	WER (%)	19.1	18.8	29.4	14.14
	SER (%)	76.6	70.71	84.68	60.18
Washington	CER (%)	6.9	24.1	18.7	5.1
	WER (%)	20.7	47.6	45.4	19.95
	SER (%)	77.8	99.4	100	68.45
Sadri_Dates	CER (%)	3.66	7.98	13.41	2.99
	WER (%)	8.06	17.08	24.12	6.67
	SER (%)	44.85	74.75	86.37	36.87

### 5.2 Comparison of the proposed architecture with Puigcerver, Flor and Bluche regarding execution time

In Table 7 and Table 8, time comparison on GPU in the proposed architecture the proposed architecture, Flor, Puigcerver, and Bluche on the datasets of IAM, Washington and Sadri\_Dates in each iteration per image in the training phase and test phase that includes image recognition and CTC decoding. As it is clear from the execution time comparison, the proposed architecture is almost 1.5 times more time-consuming in the training phase. However, it is not much different from other architectures in the testing phase.

### 5.3 Comparison of the proposed method with other methods

Compared to other approaches for recognition, as presented in Table 7, it can be observed that the proposed optical model outperforms similar models presented by various researchers using benchmark datasets.

Table 6: Recognition output by the proposed approach

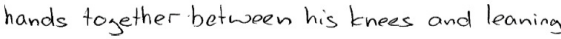
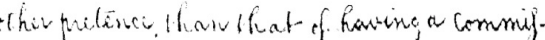
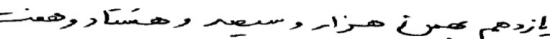
IAM	Original image:	
	Ground Truth:	hands together between his knees and leaning
	Predicted:	hands together between his knees and leaning
Washington	Original image:	
	Ground Truth:	other pretence, than that of having a commis-
	Predicted:	other pretence, than that of having a Commif-
Sadri_Dates	Original image:	
	Ground Truth:	یازدهم بهمن هزار و سیصد و هشتاد و هفت
	Predicted:	یزهم بهمن هزار و سیصد و هشتاد و هف

Table 7: Comparison of execution times on GPU across architectures and datasets in the training phase.

	DataSet	IAM	Washington	SadriDates
	<b>Architecture</b>			
<b>Flor</b>	Time per iteration (s)	9.33	100.48	136.33
	Time per image (s)	0.0189	0.0115	0.0127
<b>Puigcerver</b>	Time per iteration (s)	13.67	108.40	137.90
	Time per image (s)	0.0277	0.0125	0.0129
<b>Bluche</b>	Time per iteration (s)	2.58	52.92	74.84
	Time per image (s)	0.0052	0.0061	0.007
<b>Proposed</b>	Time per iteration (s)	15.06	197.71	235.32
	Time per image (s)	0.0306	0.0228	0.0221

Table 8: Comparison of testing phase (Image recognition and CTC decoding) execution time on GPU across architectures and datasets.

	DataSet	IAM	Washington	SadriDates
	<b>Architecture</b>			
<b>Flor</b>	Time per image (s)	65.88	68.455	67.655
<b>Puigcerver</b>	Time per image (s)	66.38	67.515	66.25
<b>Bluche</b>	Time per image (s)	64.88	63.875	65.325
<b>Proposed</b>	Time per image (s)	68.5	70.84	69.43

## 6 Conclusion and future work

In handwritten text recognition tasks, each sheet contains multiple rows which can be segmented using segmentation techniques. Due to the overlap between words in handwritten sentences, segmentation by such techniques poses problems. In handwritten sentence recognition, a gated multi-scale convolutional neural network with RNN and CTC, a classifier without segmentation, are applied. First, input images are preprocessed, and then, the CTC classifier is employed to extract features and create a proper probability matrix. Finally, the probability matrix is converted into text using the Beam Search algorithm. Some experiments are conducted to show the efficiency of the proposed approach on two well-known Latin datasets in handwritten text recognition and a Persian handwritten dataset.

Handwritten sentence recognition takes an image as an input and returns all the recognized texts from the image. To increase the precision of the output, another system called ‘‘Spell Checker’’ can be built and used in the post-processing stage, which takes a text as an input and returns corrected text. This can reduce errors and improves the overall recognition rate for Persian handwritten texts.

## Acknowledgment

The authors would like to express their sincere gratitude to Ms. Faezeh Safari, from Razi University, for her valuable comments that greatly improved the manuscript.

## References

- [1] A.A. Aburas and S.M. Rehiel, *Off-line Omni-style handwriting Arabic character recognition system based on wavelet compression*, Arab Res. Institute Sci. Eng. **3** (2007), no. 4, 123–135.
- [2] J. Aradillas, J. Murillo-Fuentes, and P. Olmos, *Boosting offline handwritten text recognition in historical documents with few labeled lines*, IEEE Access, **9** (2021), 76674–76688.
- [3] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473, (2014).
- [4] D. Banerjee, P. Bhowal, S. Malakar, E. Cuevas, M. Pérez-Cisneros, and R. Sarkar, *Z-transform-based profile matching to develop a learning-free keyword spotting method for handwritten document images*, Int. J. Comput. Intell. Syst. **15** (2022), no. 1, 93.
- [5] A. Chaudhuri, K. Mandaviya, P. Badelia, and S.K. Ghosh, *Optical Character Recognition Systems*, Springer International Publishing, 2017.
- [6] K.-N. Chen, C.-H. Chen, and C.-C. Chang, *Efficient illumination compensation techniques for text images*, Digital Signal Process. **22** (2012), no. 5, 726–733.
- [7] X. Chen, L. Jin, Y. Zhu, C. Luo, and T. Wang, *Text recognition in the wild: A survey*, ACM Comput. Surveys **54** (2021), no. 2, 1–35.
- [8] A. Chowdhury and L. Vig, *An efficient end-to-end neural model for handwritten text recognition*, arXiv preprint arXiv:1807.07965, (2018).
- [9] Y.N. Dauphin, A. Fan, M. Auli, and D. Grangier, *Language modeling with gated convolutional networks*, Int. Conf. Machine Learn., (2017), 933–941.
- [10] H. El Bahi and A. Zatni, *Text recognition in document images obtained by a smartphone based on deep convolutional and recurrent neural network*, Multimed. Tools Appl. **78** (2019), no. 18, 26453–26481.
- [11] A. Fischer, E. Indermühle, H. Bunke, G. Viehhauser, and M. Stolz, *Ground truth creation for handwriting recognition in historical documents*, Proc. 9th IAPR Int. Workshop Document Anal. Syst., 2010, pp. 3–10.
- [12] R. Geetha, T. Thilagam, and T. Padmavathy, *Effective offline handwritten text recognition model based on a sequence-to-sequence approach with CNN-RNN networks*, Neural Comput. Appl. **33** (2021), no. 17, 10923–10934.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, *Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification*, IEEE Int. Conf. Comput. Vision (ICCV), 2015, pp. 1026–1034.
- [14] S. Ioffe, *Batch renormalization: Towards reducing minibatch dependence in batch-normalized models*, Adv. Neural Inf. Process. Syst. **30** (2017).
- [15] L. Kang, P. Riba, M. Rusiñol, A. Fornés, and M. Villegas, *Pay attention to what you read: Non-recurrent handwritten text-Line recognition*, Pattern Recogn. **129** (2022), 108766.
- [16] H. Karimi, A. Esfahanimehr, M. Mosleh, F.M.J. Ghadam, S. Salehpour and O. Medhati, *Persian handwritten digit recognition using ensemble classifiers*, Proc. Comput. Sci. **73** (2015), 416–425.
- [17] B.R. Kavitha and C. Srimathi, *Benchmarking on offline handwritten Tamil character recognition using convolutional neural networks*, J. King Saud Univ.-Comput. Info. Sci. **34** (2022), no. 4, 1183–1190.
- [18] D.P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).
- [19] V. Kukreja, *A retrospective study on handwritten mathematical symbols and expressions: Classification and recognition*, Eng. Appl. Artif. Intell. **103** (2021), 104292.
- [20] A. Kumar, S. Sarkar and C. Pradhan, *Malaria disease detection using CNN technique with SGD, rmsprop and adam optimizers*, S. Dash, B. Acharya, M. Mittal, A. Abraham and A. Kelemen (eds), Deep learning techniques for biomedical and health informatics, Studies in Big Data: Springer, 2020, pp. 211–230.
- [21] U.V. Marti and H. Bunke, *The IAM-database: An English sentence database for offline handwriting recognition*, Int. J. Doc. Anal. Recog. **5** (2002), no. 1, 39–46.
- [22] S. Nasrollahi and A. Ebrahimi, *Printed Persian subword recognition using wavelet packet descriptors*, J. Eng.

- 2013** (2013), 1–11.
- [23] A.F. Neto, B.L. Bezerra, and A.H. Toselli, *Towards the natural language processing as spelling correction for offline handwritten text recognition systems*, Appl. Sci. **10** (2020), no. 21.
- [24] X. Qu, W. Wang, K. Lu, and J. Zhou, *Data augmentation and directional feature maps extraction for in-air handwritten Chinese character recognition based on convolutional neural network*, Pattern Recog. Lett. **111** (2018), 9–15.
- [25] J. Sadri, M.R. Yeganehzad, and J. Saghi, *A novel comprehensive database for offline Persian handwriting recognition*, Pattern Recog. **60** (2016), 378–393.
- [26] G. Sarker, M. Besra, and S. Dhua, *A programming based handwritten text identification*, Int. Conf. Adv. Comput. Engin. Appl., 2015, pp. 472–477.
- [27] H. Scheidl, *Handwritten text recognition in historical documents*, PhD diss., Wien, 2018.
- [28] P. Shirvani, M. Vatankhah Khouzani, and K. Yaghmaie, *Persian text recognition using n-gram language models and grammatical refinement*, JSDP **11** (2014), no. 1, 107–115.
- [29] J. Sueiras, V. Ruiz, A. Sanchez, and J.F. Velez, *Offline continuous handwriting recognition using sequence to sequence neural networks*, Neurocomputing **289** (2018), 119–128.
- [30] O. Surinta, M.F. Karaaba, L.R. Schomaker, and M.A. Wiering, *Recognition of handwritten characters using local gradient feature descriptors*, Eng. Appl. Artif. Intell. **45** (2015), 405–414.
- [31] G. Tong, Y. Li, H. Gao, H. Chen, H. Wang, and X. Yang, *MA-CRNN: A multi-scale attention CRNN for Chinese text line recognition in natural scenes*, Int. J. Document Anal. Recog. **23** (2020), no. 2, 103–114.
- [32] A. Vinciarelli and J. Luetttin, *A new normalization technique for cursive handwritten words*, Pattern Recog. Lett. **22** (2001), no. 9, 1043–1050.
- [33] X. Wang, A. Bao, Y. Cheng, and Q. Yu, *Weight-sharing multi-stage multi-scale ensemble convolutional neural network*, Int. J. Machine Learn. Cybernet. **10** (2019), no. 7, 1631–1642.
- [34] H. Wu and X. Gu, *Towards dropout training for convolutional neural networks*, Neural Networks **71** (2015), 1–10.