HTR-Flor: A Deep Learning System for Offline Handwritten Text Recognition

Arthur Flor de Sousa Neto
Escola Politécnica de Pernambuco
Universidade de Pernambuco
Recife, Brazil
afsn@ecomp.poli.br

Byron Leite Dantas Bezerra
Escola Politécnica de Pernambuco
Universidade de Pernambuco
Recife, Brazil
byron.leite@upe.br

Alejandro Héctor Toselli
Pattern Recognition and Human Language Technology
Universitat Politècnica de València
València, Spain
ahector@prhlt.upv.es

Estanislau Baptista Lima
Escola Politécnica de Pernambuco
Universidade de Pernambuco
Recife, Brazil
ebl2@ecomp.poli.br

***Abstract*—In recent years, Handwritten Text Recognition (HTR) has captured a lot of attention among the researchers of the computer vision community. Current state-of-the-art approaches for offline HTR are based on Convolutional Recurrent Neural Networks (CRNNs) excel at scene text recognition. Unfortunately, deep models such as CRNNs, Recurrent Neural Networks (RNNs) are likely to suffer from vanishing/exploding gradient problems when processing long text images, which are commonly found in scanned documents. Besides, they usually have millions of parameters which require huge amount of data, and computational resource. Recently, a new class of neural network architecture, called Gated Convolutional Neural Networks (Gated-CNN), has demonstrated potentials to complement CRNN methods in modeling. Therefore, in this paper, we present a new architecture for HTR, based on Gated-CNN, with fewer parameters and fewer layers, which is able to outperform the current state-of-the-art architectures for HTR. The experiment validates that the proposed model has statistically significant recognition results, surpassing previous HTR systems by an average of 33% over five important handwritten benchmark datasets. Moreover, the proposed model is able to achieve satisfactory recognition rates even in case of few training data. Finally, its compact architecture requires less computational resources, which can be applied for real-world applications that have hardware limitations, such as robots and smartphones.**

I. Introduction

Handwritten Text Recognition (HTR) has attracted intense attention in recent years due to its vast applications in both industrial and as an academic research topic. HTR systems have the purpose of transcribing cursive text to the digital medium (ASCII, Unicode) whether through dynamic (online) or static (offline) information [1]. Thus, images are the source of information to offline text recognition, which can be applied for transcriptions of historical manuscripts [2], medical prescriptions [3], forms [4], and so on. This emphasizes the need for research into the area of building large scale HTR systems for many languages and scripts.

Historically, offline HTR systems have been formulated as a sequence matching problem: a sequence of features extracted from input data (images) is matched to an output sequence composed of characters. During the last decade, considerable efforts to employ computer vision techniques to HTR systems have been made. Predominantly, Hidden Markov Models (HMM) [5]–[7] is one of the most popular approaches for solving the problem in HTR systems. However, HMM failed to make use of the context information, specially in a long text sequence, due to the Markovian assumption that each observation depends only on the current state.

In the last few years, Deep Learning methods, more precisely Convolutional Recurrent Neural Networks (CRNN), have demonstrated drastic improvement over traditional methods for the task of HTR. Since first introduced, CRNN for HTR has been constantly breaking state-of-the-art results and being deployed in industrial application [8]. Inside CRNN, the role of the sequence decoder is often implemented as Long Short-Term Memory (LSTM) [9]. In order to improve the accuracy for HTR, many others methods have been proposed, such as the Multidimensional LSTM (MDLSTM) [10] which extends the capability of the RNNs architectures to multidimensional data. However, the computational cost and complexity of the MDLSTM [11], [12] have led to new studies that bring simpler optical models [13], by using Bidirectional Long Short-Term Memory (BLSTM) [14]. This approach already offers results close to the known MDLSTM, such as the CNN-BLSTM and Gated-CNN-BLSTM models [15].

Despite the promising empirical results, the optical models have difficulties in remembering long contexts due to vanishing/exploding gradient problems. Additionally, these existing optical models usually have millions of trainable parameters to achieve better results, which makes them challenge to be implemented in many real-world applications [16]. On the other hand, models that have few parameters, such as Gated-CNN approaches, exchange high performance for simplicity of the model [17].

In this way, we propose a new Gated Convolutional Recurrent Neural Network (Gated-CRNN) architecture for offline HTR systems, which brings the latest machine learning techniques and approaches used in the field of Natural Language Processing, such as the Gated mechanism, presented by Dauphin [18], and the application of Bidirectional Gated Recurrent Units (BGRU) [19]. Thus, the proposed Gated-CNN-BGRU optical model involves a few parameters (thousands) and achieves a low error rate in the process of text recognition (line-level and segmentation-free). The contributions are based on the following aspects:

- Able to handle long sentences with different styles, variations and noise, even in case of limited training data.
- Improve recognition results from the CNN-BLSTM approach through the new Gated-CNN-BGRU architecture.
- Reduce the number of trainable parameters (thousands) through the Gated-CNN-BGRU architecture, making the model smaller and with lower computational cost instead of the traditional CNN-BLSTM (millions).

A variety of experiments on several well-known datasets, such as Bentham [20], IAM [21], RIMES [22], Saint Gall [23] and Washington [24], showed that the proposed model is capable of surpassing the performance of the previous models presented by [16] and [17]. Finally, an open source implementation for the reproducibility is also provided[1].

The remaining of this paper is organized as follows. In section II, reference optical models of the literature are described. Then, in section III, the proposed model is presented. In section IV, the methodology and experimental setup are explained. In section V, the experimental results obtained from the models in each dataset are discussed. Finally, section VI draws the conclusions that summarize the paper.

II. Related Works

In the HTR systems explored in this paper, the operations of a text recognition model follow three steps: (*i*) images are the inputs of the CNN layers to extract features; (*ii*) the RNN layers propagate the information from CNN and map the features in both directions of the sequence (bidirectional); and finally (*iii*) the Connectionist Temporal Classification (CTC) [25], which calculates loss value for model training and decodes into the final text for model inference. Thus, state-of-the-art optical models are presented in the following subsections.

A. Convolutional Recurrent Neural Networks

The architecture presented by Puigcerver [16] uses a traditional CRNN approach, where it has a high level of recognition rate and many parameters (around 9.6 million). The Figure 1 shows the workflow through the 5 convolutional and 5 BLSTM layers of the architecture.

The convolutional block is composed by layers with 3x3 kernels and the numbers of filters per layer following the order of 16*n* (16, 32, 48, 64, 80). MaxPooling with 2x2 kernel is applied in the first three layers and dropout (probability 0.2) in the last three to avoid overfitting [26]. In addition, Glorot uniform [27] is applied as initializer and Leaky Rectifier Linear Units (LeakyReLU) as activator [28]. Batch Normalization [29] is also used in all convolutional layers to normalize the inputs of non-linear activation functions.

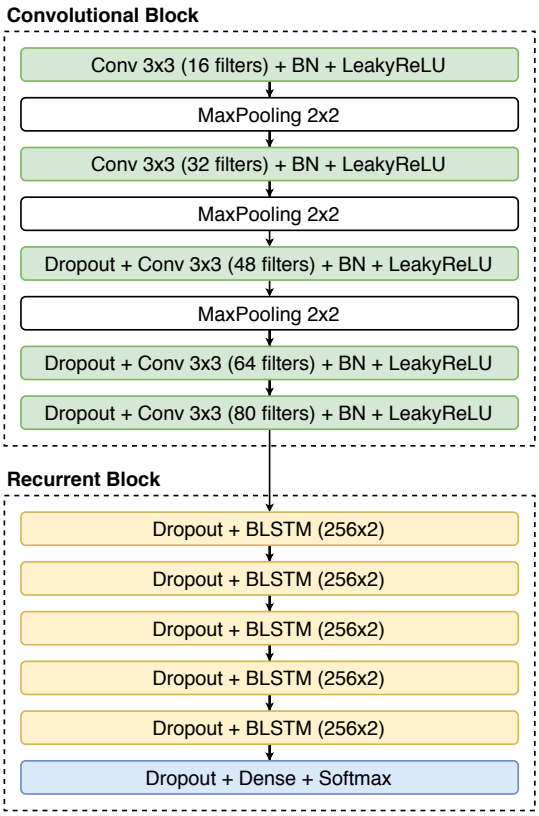

Fig. 1. Workflow of the Puigcerver architecture.

The recurrent block contains the implementation of BLSTM with dropout (probability 0.5) in the LSTM cells [30]. The number of hidden units in all LSTMs is set to 256. Finally, the model has a dense layer with a size equal to the charset size + 1 (CTC blank symbol). The dropout is also applied before the dense layer (probability 0.5).

B. Gated Convolutional Recurrent Neural Networks

The Gated-CNN approach for HTR systems, presented by Bluche and Messina [17], proposes to extract more relevant resources compared to traditional convolution. This makes the model learn better, even with few parameters to train. This gated mechanism, uses all input features (x) to perform a sigmoid activation (s) and the result is a pointwise multiplication between input (original features) and output features:

$$y = s(x) \odot x \tag{1}$$

Thus, the Gated-CNN-BLSTM architecture [17], unlike Puigcerver approach, has very few parameters (around 730 thousand), making it a compact and fast model. The Figure

[1]https://github.com/arthurflor23/handwritten-text-recognition

2 presents the Gated-CNN-BLSTM workflow through the 8 convolutional layers (3 gated included) and 2 BLSTM.

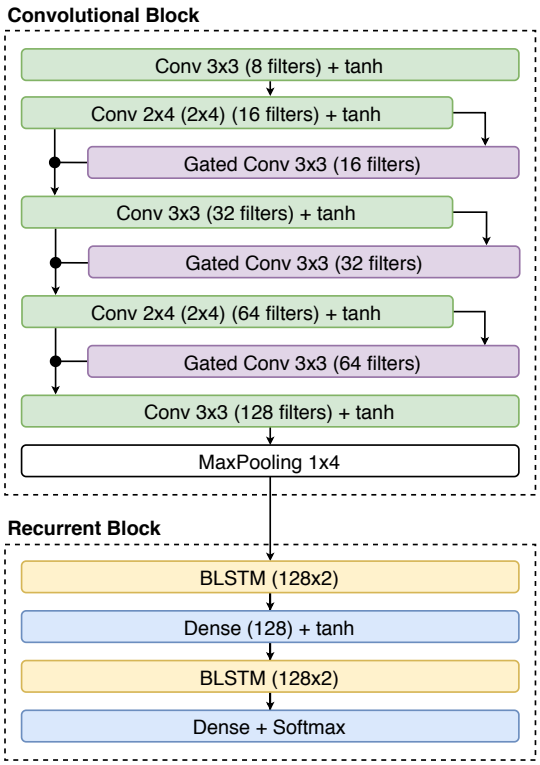

Fig. 2. Workflow of the Bluche architecture.

The convolutional block consists of mini-blocks with traditional and gated convolutions, except for the first and last layers, so: (*i*) is a 3x3 convolution (8 features); (*ii*) is a 2x4 convolution and a 3x3 gated convolution (16 features); (*iii*) a 3x3 convolution and a 3x3 gated convolution (32 features); (*iv*) a 2x4 convolution and a 3x3 gated convolution (64 features); and (*v*) a 3x3 convolution (128 features). In addition, Glorot uniform [27] is applied as initializer and Hyperbolic Tangent function (tanh) as activator.

The recurrent block contains 2 BLSTM alternated by dense layer (tanh activation). The number of hidden units in LSTMs is set to 128. Finally, the model has a dense layer with a size equal to the charset size + 1 (CTC blank symbol).

III. Proposed Model

The proposed model is inspired by [16] and [17] architectures, aiming at: (*i*) to achieve better results than the Puigcerver model; and (*ii*) to keep a low number of parameters, such as the Bluche model.

In this way, we use the Gated-CNN approach presented by Dauphin et al. [18] for the extraction of most relevant features in images. This gated mechanism has the same objective as Bluche's approach, however there is a slight difference. It uses only half of the input features (h_1) to perform sigmoid activation (s), while the other half does not (h_2), and finally, the result is a pointwise multiplication between the two halves:

$$y = s(h_1) \odot h_2 \tag{2}$$

This approach allows a better use of the Gated mechanism, in which it maintains few parameters (around 820 thousand) and a better performance of the proposed model. In addition, we also use BGRU instead of the traditional BLSTM. In the Figure 3 is presented the workflow through the 11 convolutional layers (5 gated included) and 2 BGRU.

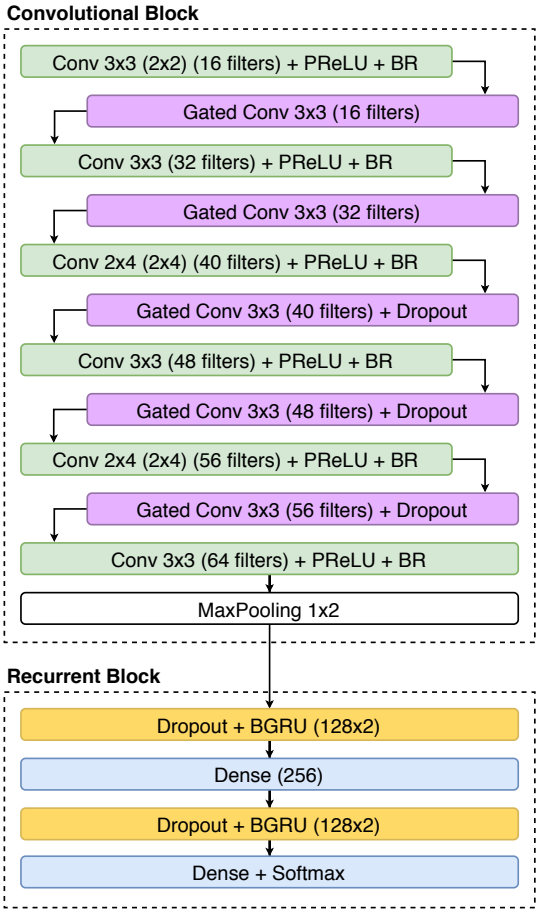

Fig. 3. Workflow of the proposed architecture.

The convolutional block consists of mini-blocks with traditional and gated convolutions, so: (*i*) has a 3x3 convolution and a 3x3 gated convolution (16 features); (*ii*) a 3x3 convolution and a 3x3 gated convolution (32 features); (*iii*) a 2x4 convolution and a 3x3 gated convolution (40 features); (*iv*) a 3x3 convolution and a 3x3 gated convolution (48 features); (*v*) a 2x4 convolution and a 3x3 gated convolution (56 features); and (*vi*) a 3x3 convolution (64 features). The He uniform is used as initializer with Parametric Rectified Linear Unit (PReLU) as activator [31]. The Batch normalization [32] is applied in all convolutional layers, followed by dropout (probability 0.2) in the last three Gated mechanisms.

The recurrent block contains 2 BGRU with dropout (probability 0.5) in the GRU cells alternated by a dense layer. The number of hidden units in GRUs is set to 128. Finally, the

model has a dense layer of size equal to the charset size + 1 (CTC blank symbol).

IV. Materials and Methods

In order to compare the proposed model with the state-of-the-art, an experimental evaluation was done using Bentham [20], IAM [21], RIMES [22], Saint Gall [23] and Washington [24] datasets, all with segmentation-free approach.

A. Datasets

The Bentham database [20] is a collection of manuscripts written by English philosopher Jeremy Bentham (1748-1832). This historical dataset, shown in Figure 4, has around 11,500 text lines and is the most complex among the five datasets adopted. It also has a considerable amount of punctuation marks in the texts.

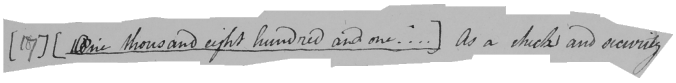

Fig. 4. Sample image from the Bentham dataset.

The Institut für Informatik und Angewandte Mathematik (IAM) database [21] contains forms with English manuscripts, which can be considered as a simple base since it has a good quality for text recognition (Figure 5). However, it brings the challenge of having several writers, that is, the cursive style is unrestricted and does not have a pattern. The amount of data has about 9,000 text lines.

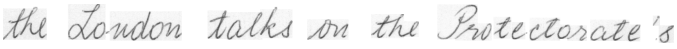

Fig. 5. Sample image from the IAM dataset.

The Reconnaissance et Indexation de données Manuscrits et de fac similÉS (Rimes) database [22] is a collection of over 12,000 text lines written in French language (Figure 6) by several writers. The text recognition is considered easy because there is a good writing of the texts, however, the French language brings accented letters challenge.

Fig. 6. Sample image from the RIMES dataset.

The Saint Gall database [23] brings manuscripts in Latin from the 9th century of only one writer (Figure 7). The images obtained are already binarized and normalized. The challenge for this collection is to deal with overfitting, since it has around 1,400 text lines in total and the writing style is very regular.

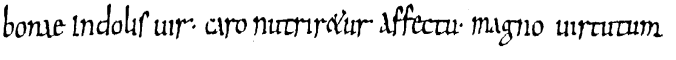

Fig. 7. Sample image from the Saint Gall dataset.

Lastly, Washington [24] was built from George Washington papers at the Library of Congress in English language from the 18th century. This set of historical manuscripts brings two writers and fewer data than Saint Gall (total of 656), in which emphasize the overfitting challenge. In addition, the images are binarized and normalized (shown in Figure 8).

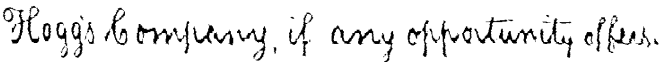

Fig. 8. Sample image from the Washington dataset.

For the data partitioning (training, validation and testing sets), the traditional standard methodology presented in each dataset work was used, except for RIMES which only has the training and testing partitions defined by default. In the RIMES case, we set the validation partition to a subset of 10% of training partition. Table I details the distribution of text lines partitions for each dataset.

TABLE I
Distribution of text lines partitions

Dataset	Training	Validation	Test	Total
Bentham	9,195	1,415	860	11,470
IAM	6,161	900	1,861	8,922
RIMES	10,193	1,133	778	12,104
Saint Gall	468	235	707	1,410
Washington	325	168	163	656

B. Experimental Setup

The optical models proposed by Puigcerver [16] and Bluche [17], used here as reference models, were evaluated following different experimental methodology according to their original works. For instance, in case of the Bluche's work, the authors included in the training set an amount of 132,000 text images of private documents. In case of Puigcerver's work, they used as input the images of entire paragraphs. In addition, for each scenario, the model was fine-tuned with its own set of specific hyperparameters (mini-batch size, learning rate, epochs to early stopping and so on).

Therefore, in order to make a fair comparison between the models and statistically validate the results from the same perspective, we followed the same methodology used by [15], in which the same workflow and hyperparameters were applied to all approaches and datasets.

In this way, we trained the optical models to minimize the validation loss value of the CTC function and get the best results. Then, we use the RMSprop optimizer [33] with the learning rate of 0.001 and mini-batches of 16 image samples per step. Reduce learning rate on plateau (factor 0.2) and Early Stopping mechanisms are also applied after 15 and 20 epochs, respectively, without improving the value of the loss of validation. It is worth mentioning that Vanilla Beam Search

algorithm [34] was used as CTC decode function in inference mode ($beam\_width = 10$). Furthermore, a common charset was used for encoding and decoding, consisting of printable and accented characters from the ASCII table (150 in total).

To improve and normalize images for all models, we applied the following preprocessing steps: first, the Illumination Compensation [35] to remove shadows and balance brightness/contrast; second, deslanting [36] to soften the cursive style; third, a resizing of 1024x128x1 (Height x Width x Channel) with padding was also done in all input images; and finally, a data augmentation increased the amount of training partition through random morphological and displacement transformations, such as rotation (up to 3 degrees), resizing (up to 5%), displacement of height and width (up to 5%), erosion (up to 5x5 kernel) and dilation (up to 3x3 kernel).

To refine the results of the optical models, we applied the Language Model through statistical characters N-grams (SRILM Toolkit[2]). This model can be efficiently trained using only plain text from the transcripts as a corpus (without the images) of each dataset under analysis [37].

Finally, all training was conducted on the Google Colab platform[3], which offers Linux operating system with 12GB memory and GPU NVIDIA Tesla P100 16GB.

C. Experimental Evaluation

The most usual evaluation metrics for HTR systems were adopted: (*i*) Character Error Rate (CER) and (*ii*) Word Error Rate (WER). These metrics are calculated through the Levenshtein Distance [38] between ground truth and predictions, for both characters and words level. As expected, the WER values tend to be greater than CER, since WER corresponds to the distribution of characters error in words [37].

For statistical testing, we conducted twenty training executions for each optical model in each dataset [39] and used Wilcoxon test [40] with 5% significance, such as adopted in [16]. As null hypothesis we considered $H_0 : \mu_1 \geq \mu_2$, and as alternative hypothesis $H_1 : \mu_1 < \mu_2$. We analyzed the hypotheses for both the CER and WER scenarios, where μ_1 is the average of the errors of the proposed model and μ_2 is the average of the errors of the other model in comparison. This means that the *p*-value must be lower than α = 0.05 to assume that the proposed model offer significantly lower error rate.

V. Results and Discussion

First, in the statistical analysis, we used the best results obtained from each dataset, considering the full text (punctuation marks included). Then, we computed CER *p*-value and WER *p*-value lower than 0.01 in all datasets. This is below the standard α = 0.05 (*p*-value < 5e-2), meaning that we can assume that the proposed model, based on Gated-CNN-BGRU, has a significantly lower CER and WER in the test partitions of each tested dataset. The *p*-values in case of each dataset are presented in brackets in the corresponding table in the following discussion.

In the Bentham dataset, the best results were obtained using the char 9-gram language model. Punctuation marks correspond to up to 25% of the error rate per word. On the other hand, considering the full text and test set, the proposed model reached CER of 3.98% with WER of 9.80%, Puigcerver 4.65% with 12.05% and Bluche 6.71% with 16.82%. Therefore, the proposed model achieved a statistically significant decrease in WER corresponding to 2.97 percentage points on Puigcerver, while 7.02 on Bluche. Table II details the results between the optical models, also considering the text without punctuation marks (only words).

TABLE II
CER and WER results in the Bentham test set

Optical Model	**Full Text**		**Only Words**	
+ char 9-gram	**CER**	**WER**	**CER**	**WER**
Puigcerver	4.65% (±0.07) [3.82e-02]	12.05% (±0.17) [5.64e-03]	3.95% (±0.06) [1.38e-02]	9.07% (±0.17) [7.21e-04]
Bluche	6.71% (±0.09) [9.04e-13]	16.82% (±0.20) [2.86e-15]	5.77% (±0.08) [1.08-e14]	13.76% (±0.21) [3.33e-17]
Flor	**3.98%** (±0.06)	**9.80%** (±0.14)	**3.33%** (±0.06)	**6.65%** (±0.13)

In the IAM dataset, the best results were obtained using the char 8-gram language model and the punctuation marks correspond only 2% of the error rate per word. In this way, also considering the full text of the test set, we obtained CER of 3.72% with WER of 11.18%, while Puigcerver 4.94% with 13.73%, and Bluche 6.60% with 17.89%. This means that the proposed model also outperforms the reference models in IAM dataset. According to Table III, it is observed a decrease in WER corresponding to 2.55 percentage points on Puigcerver and 6.71 on Bluche.

TABLE III
CER and WER results in the IAM test set

Optical Model	**Full Text**		**Only Words**	
+ char 8-gram	**CER**	**WER**	**CER**	**WER**
Puigcerver	4.94% (±0.05) [1.17e-11]	13.73% (±0.12) [1.37e-07]	4.31% (±0.04) [8.33e-11]	12.10% (±0.13) [1.65e-02]
Bluche	6.60% (±0.06) [6.88e-48]	17.89% (±0.15) [2.86e-38]	6.13% (±0.06) [1.66e-48]	17.64% (±0.16) [3.31e-33]
Flor	**3.72%** (±0.04)	**11.18%** (±0.11)	**3.37%** (±0.04)	**10.92%** (±0.12)

In the RIMES dataset, we used 12-gram language model for the best results and punctuation marks consist 14% of the error rate per word. Considering the full text of the test set,

[2]http://www.speech.sri.com/projects/srilm
[3]https://colab.research.google.com

the proposed model reached the CER of 3.27% with WER of 11.14%, Puigcerver 3.79% with 11.48% and Bluche 5.16% with 14.73%. Again, based on the *p*-values reported on Table IV, the proposed model statistically outperformed the baseline models, although it is verified a closer CER and WER of the Puigcerver.

TABLE IV
CER AND WER RESULTS IN THE RIMES TEST SET

Optical Model	**Full Text**		**Only Words**	
+ char 12-gram	**CER**	**WER**	**CER**	**WER**
Puigcerver	3.79% (±0.06) [1.03e-06]	11.48% (±0.18) [2.44e-02]	3.23% (±0.05) [1.02e-09]	9.89% (±0.18) [1.14e-05]
Bluche	5.16% (±0.07) [3.05e-41]	14.73% (±0.18) [5.43e-30]	4.78% (±0.07) [3.05e-61]	14.63% (±0.21) [9.61e-57]
Flor	**3.27%** (±0.05)	**11.14%** (±0.19)	**2.63%** (±0.04)	**8.71%** (±0.18)

The Saint Gall dataset is the only one among the others that does not have punctuation marks in the text, however, it has the longest words. In this scenario, we used the char 11-gram language model for the best results, in which the proposed model obtained CER of 5.26% with WER of 21.14%, while Puigcerver 5.95% with 23.37%, and Bluche 6.01% with 23.73%. Once, the proposed model statistically outperformed the reference models, according to the reported *p*-values in Table V.

TABLE V
CER AND WER RESULTS IN THE SAINT GALL TEST SET

Optical Model	**Full Text**	
+ char 11-gram	**CER**	**WER**
Puigcerver	5.95% (±0.03) [2.01e-06]	23.37% (±0.03) [2.23e-04]
Bluche	6.01% (±0.04) [4.96e-06]	23.73% (±0.15) [4.87e-05]
Flor	**5.26%** (±0.03)	**21.14%** (±0.13)

Finally, the Washington dataset has the least amount of data among the others. As expected, this scenario highlights the challenge of dealing with overfitting, in which it activates early stopping quickly. For this set, we used the char 10-gram language model and the punctuation marks consist only of 3% of the error rate per word. In this last dataset, we verified the largest difference in recognition rates between the proposed model and the baseline systems (Table VI). Our system outperformed significantly the reference ones, through CER of 3.01% and WER of 7.87%, while Puigcerver reached 19.29% with 32.92%, and Bluche 10.90% with 21.95%. Thus, the improvements of CER and WER were 16.28 and 25.05 percentage points, respectively, over Puigcerver model, and 7.89 and 14.08 over Bluche.

TABLE VI
CER AND WER RESULTS IN THE WASHINGTON TEST SET

Optical Model	**Full Text**		**Only Words**	
+ char 10-gram	**CER**	**WER**	**CER**	**WER**
Puigcerver	19.29% (±0.13) [1.85e-23]	32.92% (±0.20) [7.43e-22]	18.70% (±0.13) [1.46e-23]	34.26% (±0.22) [7.97e-22]
Bluche	10.90% (±0.11) [1.56e-13]	21.95% (±0.18) [1.71e-12]	10.38% (±0.11) [1.41e-14]	21.27% (±0.19) [5.29e-13]
Flor	**3.01%** (±0.04)	**7.87%** (±0.16)	**2.58%** (±0.04)	**7.59%** (±0.11)

As shown in Table VI, the differences in rates between our proposal and the state-of-the-art models selected as the baseline in this work were too much higher in this last dataset (the Washington dataset) than in the four previous experiments. Therefore, we performed one more test on the Washington dataset, but using the same parameters described in the Puigcerver [16] and Bluche [17] original works. In the Puigcerver's work was defined a learning rate of 0.0003, while the early stopping tolerance was 80 epochs, without applying the Reduce LR on Plateou. The Bluche's work defined a learning rate of 0.0004, a mini-batch of 8 image samples, the tolerance for early stopping as 80 epochs, and also without Reduce LR on Plateou. Nevertheless, the results achieved by Puigcerver and Bluche' systems with these settings were even worse (Table VII) in comparison with the ones (Table VI) achieved when these systems were trained with parameters suggested in this paper.

TABLE VII
CER AND WER RESULTS IN THE WASHINGTON TEST SET

Optical Model	**Full Text**		**Only Words**	
+ char 10-gram	**CER**	**WER**	**CER**	**WER**
Puigcerver	30.14% (±0.13) [6.31e-26]	55.62% (±0.17) [2.80e-25]	29.68% (±0.13) [5.07e-26]	58.51% (±0.20) [9.47e-26]
Bluche	34.31% (±0.12) [3.38e-26]	62.90% (±0.16) [3.34e-26]	33.90% (±0.12) [4.94e-26]	67.41% (±0.18) [7.00e-26]

To summarize all results of the experiment, we also analyzed the average error rates obtained in all datasets. Thus, the proposed model reached an average CER of 3.85% with an average WER of 12.23%. Puigcerver 7.72% with 18.71% and Bluche 7.08% with 19.02%. The increased error of Puigcerver model is due to the Washington dataset, which raises its average error rate. Figure 9 shows the average of error rate metrics of each optical model.

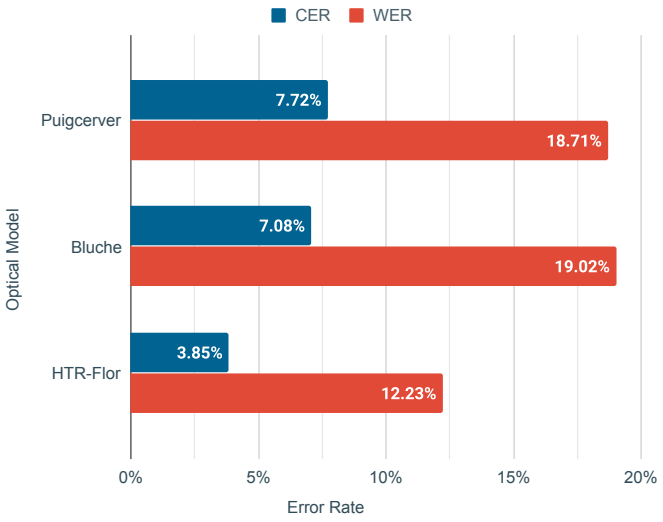

Fig. 9. Error rate summary (lower is better).

In addition, the complexity of the architecture, which impacts the size of the model and the decoding time, is another important requirement for deep neural networks. In this way, the proposed model stands out in number of trainable parameters and decoding time when compared to the Puigcerver model, but not to Bluche. However, we managed to combine the low complexity with the better recognition rate. Table VIII shows the number of parameters and the average decoding time using a standard notebook with dual core CPU (Intel i7-7500U).

TABLE VIII
NUMBER OF PARAMETERS AND AVERAGE DECODING TIME OF THE OPTICAL MODELS

Optical Model	# of params	Decoding Time
Puigcerver	9.4 M	81 ms/line
Bluche	0.7 M	32 ms/line
HTR-Flor	0.8 M	55 ms/line

Therefore, the improvements observed in the recognition rates of the proposed model, compared to the Puigcerver and Bluche approaches, can be explained mainly by the combination of: (*i*) Gated mechanism in the convolutional block; (*ii*) BGRU in the recurrent block; and (*iii*) recent deep learning techniques. In this way, we can more efficiently extract and propagate the features of the images, so that the low number of parameters and the high performance are maintained. This application is highlighted in the Washington dataset, which the proposed model achieved a significantly better result, even with the minimum volume of data.

VI. CONCLUSION

In this paper, we have presented a new Gated-CNN-BGRU architecture for offline Handwritten Text Recognition systems combined with two steps of language models.

The benchmark experiment used the same methodology for optical models under five known public datasets in the HTR field (Bentham, IAM, RIMES, Saint Gall and Washington), in which made possible the analysis from several perspectives.

The proposed model surpassed the Puigcerver and Bluche approaches, achieving an average improvement of 33% in recognition rates. Moreover, we observed the proposed model achieved very good rates even in case of small datasets, reaching up to 80% of improvement in comparison with previous works.

It is important to mention that we used hyperparameters with the focus on obtaining the best result at the lowest cost through a high learning rate and low tolerance for early stopping with reduction on plateau. Then, we could simplify the architecture with few trainable parameters (thousands), which is about 91% less than Puigcerver model.

In the future, we want to explore alternative convolutional networks to replace traditional ones, in order to further compact the model and achieve better results. We intend to carry out other evaluations in other study scenarios, such as offline handwriting recognition at the paragraph and page levels.

ACKNOWLEDGMENT

This study was financed in part by the founding public agencies: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and CNPq.

REFERENCES

[1] B. L. D. Bezerra, C. Zanchettin, A. H. Toselli, and G. Pirlo, *Handwriting: Recognition, Development and Analysis*. Nova Science Pub Inc, 07 2017.

[2] J.-A. Sánchez, V. Romero, H. A. Toselli, and E. Vidal, "Icfhr2016 competition on handwritten text recognition on the read dataset," *15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 630–635, 10 2016.

[3] E. Kamalanaban, M. Gopinath, and S. Premkumar, "Medicine box: Doctor's prescription recognition using deep machine learning," *International Journal of Engineering and Technology(UAE)*, vol. 7, pp. 114–117, 09 2018.

[4] D. Palehai and M. I. Fanany, "Handwriting recognition on form document using convolutional neural network and support vector machines (cnn-svm)," *5th International Conference on Information and Communication Technology (ICoIC7)*, 05 2017.

[5] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini, "Off-line cursive handwriting recognition using hidden markov models," *Pattern Recognition*, vol. 28, pp. 1399–1413, 09 1995.

[6] P. Doetsch, M. Kozielski, and H. Ney, "Fast and robust training of recurrent neural networks for offline handwriting recognition," *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*, vol. 2014, pp. 279–284, 12 2014.

[7] A. H. Toselli and E. Vidal, "Handwritten text recognition results on the bentham collection with improved classical n-gram-hmm methods," in *Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing (HIP@ICDAR)*, 2015.

[8] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large scale system for text detection and recognition in images," *In Proceedings of the 24th ACM SIGKDD International Conference*, pp. 71–79, 07 2018.

[9] J. S. Sepp Hochreiter, "Long short-term memory," *Neural computation*, pp. 1735–1780, 1997.

[10] A. Graves, S. Fernández, and J. Schmidhuber, "Multidimensional recurrent neural networks," *International Conference on Artificial Neural Networks*, 2007.

[11] P. Voigtlaender, P. Doetsch, and H. Ney, "Handwriting recognition with large multidimensional long short-term memory recurrent neural networks," in *15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. Shenzhen, China: IEEE, 10 2016, pp. 228–233.

[12] S. Naz, A. I. Umar, R. Ahmad, S. B. Ahmed, S. H. Shirazi, and M. I. Razzak, "Urdu nasta'liq text recognition system based on multi-dimensional recurrent neural network and statistical features," *Neural Computing and Applications*, vol. 28, no. 2, pp. 219–231, 02 2017.

[13] S. B. Ahmed, S. Naz, S. Swati, and M. I. Razzak, "Handwritten urdu character recognition using one-dimensional blstm classifier," *Neural Computing and Applications*, vol. 31, no. 4, pp. 1143–1151, 04 2019.

[14] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *Trans. Sig. Proc.*, vol. 45, no. 11, pp. 2673–2681, 11 1997.

[15] B. Moysset and R. O. Messina, "Are 2d-lstm really dead for offline text recognition?" *International Journal on Document Analysis and Recognition (IJDAR)*, pp. 1–16, 2019.

[16] J. Puigcerver, "Are multidimensional recurrent layers really necessary for handwritten text recognition?" *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pp. 67–72, 11 2017.

[17] T. Bluche and R. Messina, "Gated convolutional recurrent neural networks for multilingual handwriting recognition," *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pp. 646–651, 11 2017.

[18] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 933–941.

[19] K. Cho, B. van Merriënboer, D. Bahdanau, H. Bougares, Fethi Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 10 2014, pp. 1724–1734.

[20] B. Gatos, G. Louloudis, T. Causer, K. Grint, V. Romero, J. A. Sánchez, A. H. Toselli, and E. Vidal, "Ground-truth production in the transcriptorium project," in *11th IAPR International workshop on document analysis systems (DAS)*, 2014, pp. 237–241.

[21] U.-V. Marti and H. Bunke, "The iam-database: An english sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, pp. 39–46, 11 2002.

[22] E. Grosicki, M. Carre, J.-M. Brodin, and E. Geoffrois, "Rimes evaluation campaign for handwritten mail processing," in *ICFHR 2008 : 11th International Conference on Frontiers in Handwriting Recognition*. Montreal, Canada: Concordia University, 8 2008, pp. 1–6.

[23] A. Fischer, E. Indermühle, H. Bunke, G. Viehhauser, and M. Stolz, "Ground truth creation for handwriting recognition in historical documents," *ACM International Conference Proceeding Series*, pp. 3–10, 2010.

[24] A. Fischer, V. Frinken, A. Fornés, and H. Bunke, "Transcription alignment of latin manuscripts using hidden markov models," in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, ser. HIP'11. New York, NY, USA: Association for Computing Machinery, 2011, p. 29–36.

[25] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML - Proceedings of the 23rd International Conference on Machine Learning*, 01 2006, pp. 369–376.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 06 2014.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10)*. New Jersey, USA: Society for Artificial Intelligence and Statistics, 2010.

[28] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 448–456.

[30] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, 9 2014, pp. 285–290.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 12 2015, pp. 1026–1034.

[32] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 1942–1950.

[33] T. Tieleman and G. Hinton, "Lecture 6.5–rmsprop: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Networks for Machine Learning, 2012.

[34] K. Hwang and W. Sung, "Character-level incremental speech recognition with recurrent neural networks," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3 2016.

[35] K.-N. Chen, C.-H. Chen, and C.-C. Chang, "Efficient illumination compensation techniques for text images," *Digital Signal Processing*, vol. 22, no. 5, pp. 726–733, 2012.

[36] A. Vinciarelli and J. Luettin, "A new normalization technique for cursive handwritten words," *Pattern Recognition Letters*, vol. 22(9), pp. 1043–1050, 07 2001.

[37] J. A. Sánchez, V. Romero, A. H. Toselli, M. Villegas, and E. Vidal, "A set of benchmarks for handwritten text recognition on historical documents," *Pattern Recognition*, vol. 94, pp. 122–134, 2019.

[38] V. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics Doklady*, vol. 10, p. 707, 1966.

[39] W. J. Conover, *Practical Nonparametric Statistics*. New York: John Wiley & Sons, 1971.

[40] F. Wilcoxon, *Individual Comparisons by Ranking Methods*. New York: Springer New York, 1992, pp. 196–202.