Adaptive Context-aware Reinforced Agent for Handwritten Text Recognition

Liangke Gui
liangkeg@cs.cmu.edu
Xiaodan Liang
xiaodan1@cs.cmu.edu
Xiaojun Chang
uqxchan1@cs.cmu.edu
Alexander G. Hauptmann
alex@cs.cmu.edu

School of Computer Science
Carnegie Mellon University
PA, USA

Abstract

Handwritten text recognition has been a ubiquitous research problem in the field of computer vision. Most existing approaches focus on the recognition of handwritten words without considering the cursive nature and significant differences in the writing of individuals. In this paper, we address these problems by leveraging an adaptive context-aware reinforced agent which learns the actions to determine the scales of context regions during inference. We formulate our approach in a reinforcement learning framework. Specifically, we construct the action set with a number of context lengths. Given an image feature sequence, our model is trained to adaptively choose the optimal context length according to the observed state. An attention mechanism is then used to selectively attend the context region. Our model can generalize well from recognizing isolated words to recognizing individual lines of text while remain low computation overheads. We conduct extensive experiments on three large-scale handwritten text recognition datasets. The experimental results show that our proposed model is superior to the state-of-the-art alternatives.

1 Introduction

Handwritten text recognition (HWR) is commonly used to extract natural languages from images. It remains an open research problem, in which noisy, real-valued input streams are annotated with strings of discrete labels, such as letters or words. Handwritten text recognition presents relevant applications such as bank check reading, mail sorting, and content preservation of historical documents. Due to the importance of these applications, it has attracted increasing research attention in recent years.

Despite recent advances in scene text recognition [7, 10, 30, 44, 47], recognizing handwritten text, due to the cursive nature of handwritten characters and significant differences in the writing of individuals, remains challenging. Several attempts using convolutional neural networks (CNNs) [3, 27, 41] have been shown to produce impressively low error

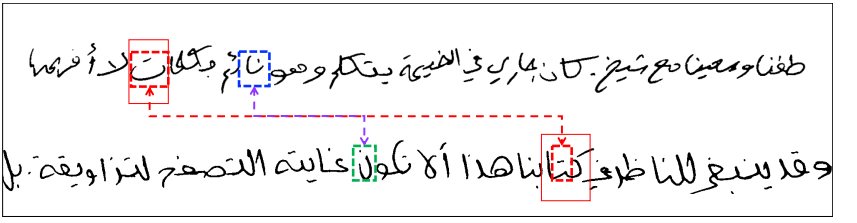

Figure 1: Two samples from KHATT dataset. The shape of the same character ت (red) varies under different surrounding context, while two different words of نا (blue) and ن (green) share a similar shape. Correctly inferring a character depends on its correlated characters which we denote as **local context**. We refer to the number of adjacent characters needed to make an inference as **context length.**

rates on handwritten word datasets. However, these systems use fixed-size CNNs and focus on isolated words which are rarely readily available in real world applications. Another general approach is to use recurrent neural networks (RNNs) associated with connectionist temporal classification (CTC) [18]. They are capable of recognizing a line of text without word-level segmentation. Doetsch *et al*. [15] use a stacked bidirectional long short-term memory (BLSTM) [19, 22] with PCA-based features. In a recent German handwritten text recognition competition [43], the top methods use architectures which generally consist of CNNs and RNNs and achieve remarkable performance. Bluche *et al*. [6] propose a MDLSTM-attention system to recognize handwritten text from paragraphs by incorporating multi-dimensional LSTM [17] and attention mechanism. We are inspired by this idea but propose significant modifications.

One observation is that the reading order of characters is typically established by convention (*e.g*. a primary order from right to left in Arabic scripts). Therefore, while LSTM is capable of capturing long-term dependencies in the handwritten text recognition task, the local context around a target position is informative to determine a character, as illustrated in Fig. 1. Characters may rely on different scales of context region. For example, due to the cursive writing, inferring the character in dash-line box may rely on the context in solid-line box. Meanwhile, within the context region, the characters may contribute differently to the inference. Motivated by this, we propose to introduce an adaptive context length selection and soft attention mechanism into the handwritten text recognition task.

To address the above mentioned issues, we present a framework that treats context regions localization as a decision making process, by which an agent would adaptively select a context length according to the observed states. In our framework, we prepare a number of context lengths as the action set. Choosing the context length is formulated as a reinforcement learning framework. By applying a policy network, an agent learns to select the optimal length of context region by analyzing the observed content. To keep the policy execution lightweight, we take all the decisions in a single step which can be seen as an instantiation of associative reinforcement learning [46]. Thus we maximize the negative loss as the global reward of our policy network.

We refer the proposed framework as Adaptive Context-aware Reinforced Agent. Our contributions are summarized as follows:

- We make the first attempt to address the handwritten text recognition problem in a

reinforcement learning framework. By learning an adaptive context-aware reinforced agent, our proposed model is capable of selectively attending context regions during inference.
- Unlike previous work on Arabic words recognition, we solve a more challenging task of Arabic handwritten text line recognition.
- We show that our proposed model generalizes well from isolated words to text lines recognition and achieves the state-of-the-art performances on several benchmarks.

Our paper is structured as follows. We first overview the recent research on handwritten text recognition, attention mechanism, and reinforcement learning in Section 2. We then present our model in Section 3, followed by a description of experiments in Section 4 and results in Section 4.2. We conclude and present future directions in Section 5.

2 Related Work

We first discuss widely used approaches for handwritten text recognition. We then discuss the recent advances in attention mechanisms and reinforcement learning which our work builds on.

Handwritten Text Recognition. Traditional approaches to handwritten text recognition are mainly focused on two key elements: the strategy to extract features and the way to decode the output of the classifiers to predict the sequence of characters [45]. Poznanski *et al*. [41] propose a CNN-N-Gram model to estimate the n-gram frequency profile given a handwritten word image. Despite of the remarkable performance on several handwritten benchmarks, the manually defined N-gram CNN model has a large number of output nodes which increases the training complexity. Shi *et al*. [44] propose a CRNN model to recognize text in the wild and is closely related to our work. In their work, a CNN model is used to extract feature sequences from input images and a recurrent network is built for making prediction for each frame of the feature sequence. While their approach is designed for scene word recognition with a constrained image scale, our model is focused on handwritten text recognition and can generalize from single word to text lines.

Attention Model. "Attention-based" methods have shown to be successful for machine translation [2], image caption generation [11, 50] and speech recognition [9, 12]. Attention-based mechanisms can allow the model to learn alignments between different modalities.Many researchers have explored different attention methods to solve the image-based text recognition task. Deng *et al*. [14] propose a coarse-to-fine attention mechanism to convert images into presentational markup by constructing a sparse coarse attention to reduce the number of fine attention cells. To recognize the text in the wild, Lee *et al*. [29] propose a R2AM model to selectively exploit image features in a coordinated way by incorporating soft attention [51]. Bluche *et al*. [6] propose a multi-dimensional LSTM architecture associated with an attention mechanism to recognize handwritten text in paragraphs without explicit segmentation. Different from previous work, we follow the idea of *local* attention [32] which can be viewed as a blend between hard and soft attention. Our model focuses on the local context around the target states and avoids the expensive computation incurred in the soft attention. Thus, our model is scalable to images with long character sequences. Mnish *et al*. [37] proposes a recurrent neural network to extract information from an image or video by adaptively selecting a sequence of regions or locations. Different from their work, we focus on handwritten text recognition and attend different regions during training and inference.

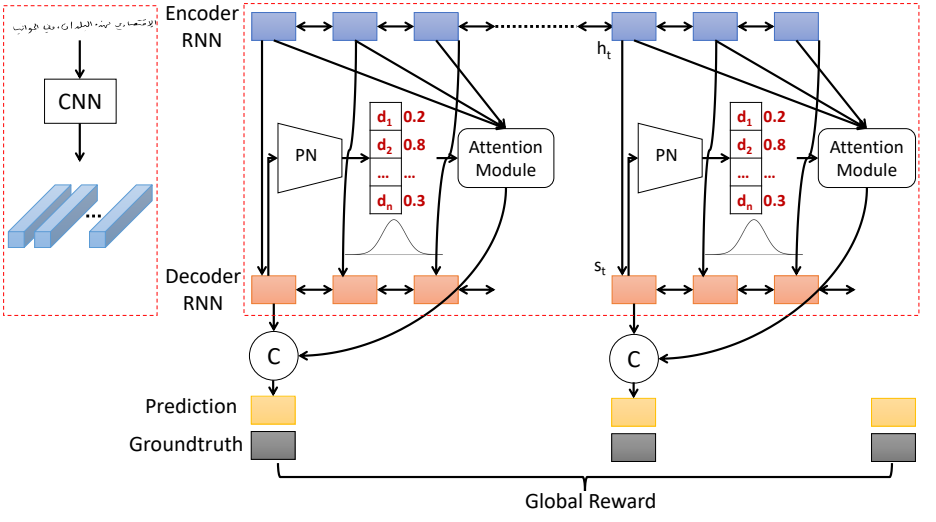

Figure 2: The framework of our proposed model. The policy network (PN) is trained to choose an optimal context length from the action set according to the observed state. The attention module then selectively attends to this context region and explicitly encode it into the local context. The context captured by LSTM and the local context are simultaneously taken into consideration during inference.

Reinforcement Learning. Reinforcement learning (RL) is to learn a policy network that determines certain actions under particular states. It is effective to optimize the sequential decision problems. Recently, several attempts have applied RL to computer vision tasks [8, 23, 24, 31, 49, 51]. Zhao *et al*. [51] and Wu *et al*. [49] explore deep RL to dynamically choose layers of CNNs during inference.A video object segmentation model [24] is proposed to learn object foreground-context regions by incorporating a reinforcement cutting-agent learning framework. In our work, we adopt a policy network to select context regions to attend according to the observed states during inference. Inspired by the BlockDrop model [49], we view our decision making process as an instantiation of associative reinforcement learning where all the decisions are taken in a single step.

3 Methodology

In offline handwritten text recognition tasks, the goal is to build a system which, given an image, produces a prediction of the image transcription. Our insight is that it is beneficial to simultaneously leverage both local context (as illustrated in Fig. 1) and global context. The key idea is that we adaptively select context region to attend during inference according to the observed states. Fig. 2 shows an overview of our framework.

Formally, given a dataset $S = \{(I,z)\}$, I is an image and z is the textual transcription. We take a raw image as input and encode it into a feature sequence s, where s_t is the state at time-step t. We train an adaptive context-aware reinforced agent to predict the context length of s_t. We then derive the expectation c_t within the window size by leveraging the soft attention mechanism. c_t is applied as the adaptive local context during inference. Details of

the model are demonstrated in the following sections.

3.1 Visual Features Encoder

The visual features of an image are extracted from a fully convolutional neural network which consists of max-pooling layers. We model it using the CNN network [44] for OCR from images (Specification is given in Table 1). The network takes the raw inputs and produces feature maps that are robust and contain high-level descriptions of the input images. Suppose the feature maps are of size $D \times H \times W$, where D denotes the number of channels and H and W are the height and width of the feature maps.

According to the translation invariance property of CNN, each column of the feature maps corresponds to a local image region as the receptive field. The feature maps are then flattened into a sequence with a length of W, each of which has $D \times H$ dimensions. Specifically, each feature vector of the feature sequence is generated from left to right on the feature maps by column. We denote the the visual feature sequence as $v = (v_1, \ldots, v_W)$. We follow the same settings [44], and fix the height of each column H as a single pixel.

Restricted by the sizes of the receptive fields, the feature sequence leverages limited image contexts. We run a RNN over the feature sequence V to model the long-term dependencies within the sequence. Formally, a RNN is a parameterized function that recursively maps an input vector and a hidden state to a new hidden state. At time t, the hidden state is updated with an input v_t in the following manner: $h_t = RNN(h_{t-1}, v_t; \theta)$. For simplicity we will describe the model as a RNN, but all experiments use the BLSTM. We denote the encoded states from v as $h = h_1, \ldots, h_W$.

Conv	MaxPool	Conv	MaxPool	Conv	Conv	MaxPool	Conv	Conv	MaxPool	Conv
3×3	2×2	3×3	2×2	3×3	3×3	2×2	3×3	3×3	2×2	2×2
num: 64		num: 128		num: 256	num: 256		num: 512	num: 512		512
sh:1 sw:1	sh:2 sw:2	sh:1 sw:1	sh:2 sw:2	sh:1 sw:1	sh:1 sw:1	sh:2 sw:1	sh:1 sw:1	sh:1 sw:1	sh:2 sw:1	sh:1 sw:1
ph:1 pw:1	ph:0 pw:0	ph:1 pw:1	ph:0 pw:0	ph:1 pw:1	ph:1 pw:1	ph:0 pw:1	ph:1 pw:1	ph:1 pw:1	ph:0 pw:1	ph:0 pw:0

Table 1: The CNN architecture configuration.

3.2 Context Features Decoder

Considering the cursive and imprecise nature in the handwritten text recognition problem, our insight is that explicitly encoded local context (as illustrated in Fig. 1) is complementary to global context when determining observed states into characters. Given a feature sequence, learning the context region localization agent would result in a nearly continuous decision-making process. To simplify this problem, we discretize the context regions into an action set and leverage a policy network to make decisions in selecting appropriate context regions.

We introduce an adaptive context-aware agent to select and attend different context regions given states at different time-steps. We first leverage a BLSTM to extract higher level of abstractions from the encoder outputs s as $s_t = RNN(s_{t-1}, h_t; \theta)$.

Adaptive Context-aware Reinforced Agent. Our method is based on Q-learning, a kind of reinforcement learning, which focuses on how an agent ought to take actions so as to maximize the final reward. The Q-learning model consists of an *agent*, *states* and a set of *actions*.

We adopt s as the sequence *states*. The searching action set $\mathcal{A}$ contains different context lengths and is denoted as $\mathcal{A} = \{d_1, \ldots, d_n\}$, where n is the number of context lengths. For an input s_t, we design a policy network to learn the expected adaptive context-aware reinforced agent, which determines the action policy $a(s_t)$ according to the observed s_t. Both the *state* and *action* are finite and discrete to ensure a relatively small searching space. Given a $(s_t, a(s_t))$, we adopt the negative loss defined in Sec. 3.3 as our reward. Following the training strategy [49], we train the policy network to predict *all actions at once* which is different from the standard reinforcement learning algorithms and is essentially a single-step Markov Decision Process (MDP) given the input states. This can also be viewed as contextual bandit [28] or associative reinforcement learning [46].

Formally, given a sequence s, we define an action policy as a multinomial distribution:

$$\pi_W(a|s) = \prod_{t=1}^{T} p_t^{a_t}, \tag{1}$$

$$p = f_{pn}(s; W), \tag{2}$$

where f_{pn} denotes the *policy network* parameterized by weights W and p is the output of the network after the softmax function. We denote the probability of the corresponding action a_t at time-step t as $p_t^{a_t}$. To learn the optimal parameters of the policy network, we maximize the following expected reward:

$$J = \mathbb{E}_{a \sim \pi_W}[R(a)]. \tag{3}$$

To maximize Eqn. 3, we utilize policy gradient [46], one of the seminal policy search methods [13], to compute the gradients of J. The gradients can be derived as:

$$\nabla_W J = E[R(a) \nabla_W \log \pi_W(a|s)], \tag{4}$$

Where W denotes the parameters of the policy network. We approximate the expected gradient in Eqn. 4 with Monte-Carlo sampling using all samples in a mini-batch. To reduce variance [46] in these gradient estimates, we utilize a self-critical baseline $R(\tilde{u})$ as in [42] and Eqn. 4 can thus be rewritten as:

$$\nabla_W J = E[(R(a) - R(\tilde{a})) \nabla_W \log \pi_W(a|s)], \tag{5}$$

where $\tilde{a}$ is defined as the maximally probable configuration under the current policy. For example, $\tilde{a}$ is the action from $\mathcal{A}$ with the index of $argmax(p_t)$.

To further encourage exploration in policy searches, we adopt a parameter α to bound the distribution p and prevent it from saturating. The modified distribution p' can be formulated as:

$$p' = \alpha \cdot p + (1 - \alpha) \cdot (1 - p). \tag{6}$$

The modified distribution p' is applied when we sample the action policies.

Local Attention. Since not every time-step of the sequence is relevant for the prediction, the model should extract the salient parts. Our local attention mechanism selectively focuses on a small window of context. In concrete details, given a predicted window size D at time-step t, the source hidden states within the window are denoted as $h_{[t-\frac{D}{2}:t+\frac{D}{2}]}$. We follow past empirical work [32] and compute the attention weight vector as:

$$a^{att} = softmax(s_t^T W_a h_{[t-\frac{D}{2}:t+\frac{D}{2}]}), \tag{7}$$

where W_a is the projection vector which will be jointly trained with the model. Then the context at time-step t is defined as an expectation of s within the window of $[t-\frac{D}{2}:t+\frac{D}{2}]$:

$$c_t = \sum_i a_i^{att} h_i. \tag{8}$$

To take both the global context and explicitly encoded local context into consideration, we use the concatenation of s_t and c_t as the representation at time-step t.

In summary, our model works as follows: f_{pn} is used to decide which window size to attend conditioned on the input feature sequence. A prediction is generated by running a forward pass and we aim to maximize the total expected reward, or equivalently minimize the negative expected reward as our loss.

3.3 Transcription Layer

Transcription is a process of converting the per-frame predictions made by the decoder module into a label sequence. Mathematically, transcription procedure is to find the label sequence with the highest probability conditioned on the per-frame predictions.

In this section, We adopt Connectionist Temporal Classification (CTC) [18] layer to transform variable-width feature tensor into a conditional probability distribution over label sequence. The probability ignores the position where each per-frame prediction is located and avoids the labor of labeling positions of individual characters.

Formally, let $\mathcal{L}$ be the alphabet and $\hat{\mathcal{L}} = \mathcal{L} \cup \{-\}$ where $-$ is a blank character. Given an input image I, the generated predictions $\pi = \{\pi_1, \ldots, \pi_T\}$, where T is the sequence length and $\pi \in \mathcal{R}^{\hat{\mathcal{L}}}$. The probability distribution over the alphabet $\hat{\mathcal{L}}$ is denoted as $y = \{y_1, ..., y_T\}$. We denote $y^t_{\pi_t}$ as the probability of generating label π_t at time-step t. The sequence π may contain blank characters and repeated labels. CTC defines a map function $\mathcal{B}$ which maps π to a concise representation l by removing blank characters and repeated labels (*e.g.* hhee–lllo–=hello).

Thus, the probability of π is defined as $p(\pi|y) = \prod_{t=1}^{T} y^t_{\pi_t}$. The conditional probability of observing the output sequence l is then given as:

$$p(l|y) = \sum_{\pi:\mathcal{B}(\pi)=l} \log p(\pi|y). \tag{9}$$

Due to the exponentially large number of summation items, directly computing Eqn. 9 is computationally infeasible. While Eqn. 9 can be efficiently computed using the forward-backward algorithm [18].

4 Experiments

4.1 Experimental Setup

In this section, we present our experiment setups by introducing the benchmarks, the experiment settings and evaluation metrics used for evaluation.

Datasets. We present results on the commonly used handwritten text recognition benchmarks. The datasets used are KHATT, IAM and RIMES, which contain images of handwritten Arabic, English and French, respectively. We use the same network for all experiments

and no language specific information is needed except for the character set of each benchmark. A brief description of these benchmarks is as follows.

The KHATT [33] database is an offline handwritten text recognition database of cursive Arabic text documents. It contains 2,000 paragraphs by 1,000 writers. The paragraphs are segmented into a total number of 9,327 lines. The database is provided with line level annotations and a standard data set splits.

The IAM [34] database is a handwritten text recognition database of mostly cursive English text documents. The training set comprises 747 documents (6,482 lines, 55,081 words), the validation set 116 documents (976 lines, 8,895 words) and the test set 336 documents (2,915 lines, 25,920 words). The texts in this database typically contain 50 characters per line.

The RIMES [21] database contains more than 60,000 words written by over 1,000 authors in French. This database has several versions with each one a super-set of the previous one. We use the latest version presented in a ICDAR 2011 contest for our experiments.

Experiment settings. We follow the lexicon-based methods [1, 4, 16, 41] and use all the dataset words, both train and test sets, as the lexicon. The model's predictions are compared with the actual image transcriptions. To ease comparison to other algorithms, we report using the same measure commonly used in the respected benchmarks. On IAM and RIMES, we show our results using WER and CER measures. Whereas on KHATT, images are annotated at line level which makes the measure of WER infeasible. We report our results using CER calculated at sequence level.

Different character sets are used for the benchmarks. More specifically, the character set for IAM contains the lower and upper case Latin alphabet. Digits are not included as they are rarely used in this dataset. For RIMES, the character set contains the lower and upper case Latin alphabet, digits and accented letters. For KHATT, as the images are at line level, the character set contains the Arabic alphabet, comma, dot, space and unknown letters.

Evaluation protocols. We apply our model to the test set and compare the predicted transcription with the ground truth transcriptions. The performance can be measured by Word Error Rate (WER) and Character Error Rate (CER). WER is the ratio of the reading mistakes calculated at the word level. CER measures the Levenshtein distance normalized by the length of the ground-truth word. That is, we measure the total number of substitutions, insertions and deletions that would be required to turn the prediction sequence into the ground-truth one.

Implementation details. In our experiments, we binarize images by applying Otsu's method [38]. The heights of images are scaled to 32 and the widths are proportionally scaled with heights. The size of hidden states for encoder and decoder modules are set as 128. We implement the neural network using PyTorch. Parameter optimization is performed using the Adam algorithm [25] with a batch size of 32 and a learning rate of 0.01. To reduce the effects of "gradient exploding", we use a gradient clipping of 0.1 [39]. We insert batch normalization layer after each convolutional layer to accelerate the training process. We empirically set values of actions as $\mathcal{A} = \{1, 5, 10, 15, 20\}$. Training the network takes around 20min on KHATT dataset using a single GPU TITAN X.

4.2 Results and Discussion

To evaluate the effectiveness of our proposed algorithm, we conduct an extensive set of experiments on handwritten words recognition benchmarks. We also investigate the ablation studies on handwritten text lines recognition benchmarks.

Handwritten word recognition task. We compare to the state of the art on IAM and RIMES datasets in Table 2. Our model outperforms previous work by large margins on the handwritten words recognition benchmarks. Wigington *et al.* [43] reports two results with/without data augmentation techniques on the test set. For a fair comparison, we compare the performance under the same experiment settings by leveraging the training set only. As Shi *et al.* [44] is closely related to our work, we report the performance on two benchmarks. While their work is focused on scene text recognition, it is still competitive compared to other previous work. Our model outperforms Shi *et al.* [44] which indicates the adaptive context-aware reinforced agent can help with recognizing handwritten words.

Database	IAM		RIMES	
Model	WER	CER	WER	CER
Boquera *et al.* [16]	15.50	6.90	-	-
Telecom ParisTech [20]	-	-	24.88	-
IRISA [20]	-	-	21.41	-
Jouve [20]	-	-	12.53	-
Kozielski *et al.* [26]	13.30	5.10	13.70	4.60
Almazan *et al.* [1]	20.01	11.27	-	-
Messina and Kermorvant [36]	19.40	-	13.30	-
Pham *et al.* [40]	13.60	5.10	12.30	3.30
Bluche *et al.* [4]	20.50	-	9.2	-
Doetsch *et al.* [15]	12.20	4.70	12.90	4.30
Bluche *et al.* [5]	11.90	4.90	11.80	3.70
Shi *et al.* [44]	6.74	3.75	4.23	2.10
Menasri *et al.* (combined) [35]	-	-	4.75	-
Poznanski *et al.* [41]	6.45	3.44	3.90	1.90
Wigington *et al.* [43]	7.18	3.93	3.84	1.82
Our work	**5.45**	**3.10**	**2.97**	**1.45**

Table 2: Comparison to previous methods on IAM and RIMES (ICDAR2011) datasets. Our model achieves the state-of-the-art performance by large margins on both benchmarks. All numbers are in percent.

Handwritten line recognition task. To test the scalability to long sequences (*e.g.* 60 characters per sequence in KHATT dataset), we compare our model to the state-of-the-art algorithms on IAM and KHATT benchmarks. Our models are trained and evaluated using full lines. The comparisons are as shown in Table 3. We report the performance of Shi *et al.*'s work [44], as it is closely related to our work and can be viewed as a baseline. Our model lowers the error rate by 1.7% compared to the baseline model. On IAM dataset, we compare our model to Bluche *et al.*'s work which achieves remarkable performance on multi-line handwritten recognition [5]. Our model outperforms their work on both line and isolated word recognition.

Ablation studies. To investigate the impact of our proposed model, we conduct an extensive set of experiments. The first experiment is to validate if local attention mechanism outperforms global attention over the full sequence. As shown in Table 3, the global attention performs worse on both benchmarks. One possible reason is that unlike other tasks (*e.g.* machine translation), global attention introduces more noise when dealing with long sequences due to the imprecise nature of handwriting. We then replace the adaptive context-aware

reinforced agent with a single fixed-size. The window size is empirically set as 9, the median value of our action sets. This modified model performs better than the baseline while consistently worse than our proposed model on both benchmarks.

Database	IAM	KHATT
Model	CER	CER
Shi *et al*. [44]	6.20	8.65
Bluche *et al*. (w/o attention) [6]	6.60	-
Bluche *et al*. (w/ attention) [6]	7.00	-
Our work (w/ GA)	8.35	10.20
Our work (w/ fixed-size LA)	5.91	7.62
Our work (full model)	**5.15**	**6.93**

Table 3: Comparison to previous methods and ablation studies on IAM and KHATT datasets. Our experiments are conducted on full lines instead of isolated words. All numbers are in percent. GA: global attention, LA: local attention.

5 Conclusion

In this paper, we have made a pioneer effort to formulate handwritten text recognition in a reinforcement learning framework and propose a novel adaptive context-aware reinforced agent to tackle this problem. The proposed method can generalize well from isolated word recognition to full lines recognition. Comprehensive experiments on commonly used benchmark datasets demonstrate the effectiveness of the proposed method. In the future, we plan to extend this method to multi-lines and paragraphs recognition without pre-segmentation.

Acknowledgement

We thank the Center for Human Rights Science of Carnegie Mellon University and the MacArthur Foundation for their support.

References

[1] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. Word spotting and recognition with embedded attributes. *TPAMI*, 36(12):2552–2566, 2014.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

[3] Théodore Bluche, Hermann Ney, and Christopher Kermorvant. Feature extraction with convolutional neural networks for handwritten word recognition. In *ICDAR*, 2013.

[4] Theodore Bluche, Hermann Ney, and Christopher Kermorvant. Tandem hmm with convolutional neural network for handwritten word recognition. In *ICASSP*, 2013.

[5] Théodore Bluche, Hermann Ney, and Christopher Kermorvant. A comparison of sequence-trained deep neural networks and recurrent neural networks optical modeling for handwriting recognition. In *SLSP*, 2014.

[6] Théodore Bluche, Jérôome Louradour, and Ronaldo Messina. Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention. In *NIPS*, 2016.

[7] Michal Bušta, Lukáš Neumann, and Jirı Matas. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *ICCV*, 2017.

[8] Qingxing Cao, Liang Lin, Yukai Shi, Xiaodan Liang, and Guanbin Li. Attention-aware face hallucination via deep reinforcement learning. In *CVPR*, 2017.

[9] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *ICASSP*, 2016.

[10] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In *ICCV*, 2017.

[11] Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. Describing multimedia content using attention-based encoder-decoder networks. *TMM*, 17(11):1875–1886, 2015.

[12] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *NIPS*, 2015.

[13] Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

[14] Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, and Alexander M Rush. Image-to-markup generation with coarse-to-fine attention. *ICML*, 2017.

[15] Patrick Doetsch, Michal Kozielski, and Hermann Ney. Fast and robust training of recurrent neural networks for offline handwriting recognition. In *ICFHR*, 2014.

[16] Salvador Espana-Boquera, Maria Jose Castro-Bleda, Jorge Gorbe-Moya, and Francisco Zamora-Martinez. Improving offline handwritten text recognition with hybrid hmm/ann models. *TPAMI*, 33(4):767–779, 2011.

[17] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *NIPS*, 2009.

[18] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006.

[19] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, 2013.

[20] Emmanuele Grosicki and Haikal El-Abed. Icdar 2011-french handwriting recognition competition. In *ICDAR*, 2011.

[21] Emmanuèle Grosicki, Matthieu Carre, Jean-Marie Brodin, and Edouard Geoffrois. Rimes evaluation campaign for handwritten mail processing. In *ICFHR*, 2008.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[23] Zequn Jie, Xiaodan Liang, Jiashi Feng, Xiaojie Jin, Wen Lu, and Shuicheng Yan. Tree-structured reinforcement learning for sequential object localization. In *NIPS*, 2016.

[24] Han Junwei, Yang Le, Zhang Dingwen, Chang Xiaojun, and Liang Xiaodan. Reinforcement cutting-agent learning for video object segmentation. In *CVPR*, 2018.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015.

[26] Michał Kozielski, Patrick Doetsch, and Hermann Ney. Improvements in rwth's system for off-line handwriting recognition. In *ICDAR*, 2013.

[27] Praveen Krishnan, Kartik Dutta, and CV Jawahar. Deep feature embedding for accurate recognition and retrieval of handwritten text. In *ICFHR*, 2016.

[28] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *NIPS*, 2008.

[29] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *CVPR*, 2016.

[30] Hui Li, Peng Wang, and Chunhua Shen. Towards end-to-end text spotting with convolutional recurrent neural networks. In *ICCV*, 2017.

[31] Xiaodan Liang, Lisa Lee, and Eric P Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *CVPR*, 2017.

[32] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *EMNLP*, 2015.

[33] Sabri A Mahmoud, Irfan Ahmad, Mohammad Alshayeb, Wasfi G Al-Khatib, Mohammad Tanvir Parvez, Gernot A Fink, Volker Märgner, and Haikal El Abed. Khatt: Arabic offline handwritten text database. In *ICFHR*, 2012.

[34] U-V Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *IJDAR*, 5(1):39–46, 2002.

[35] Farès Menasri, Jérôme Louradour, Anne-Laure Bianne-Bernard, and Christopher Kermorvant. The a2ia french handwriting recognition system at the rimes-icdar2011 competition. In *Document Recognition and Retrieval XIX*, volume 8297, page 82970Y. International Society for Optics and Photonics, 2012.

[36] Ronaldo Messina and Christopher Kermorvant. Over-generative finite state transducer n-gram for out-of-vocabulary word recognition. In *DAS Workshop*, 2014.

[37] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *NIPS*, 2014.

[38] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[39] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013.

[40] Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. Dropout improves recurrent neural networks for handwriting recognition. In *ICFHR*, 2014.

[41] Arik Poznanski and Lior Wolf. Cnn-n-gram for handwriting word recognition. In *CVPR*, 2016.

[42] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *CVPR*, 2017.

[43] Joan Andreu Sanchez, Veronica Romero, Alejandro H Toselli, and Enrique Vidal. Icfhr2016 competition on handwritten text recognition on the read dataset. In *ICFHR*, 2016.

[44] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *TPAMI*, 39(11):2298–2304, 2017.

[45] Jorge Sueiras, Victoria Ruiz, Angel Sanchez, and Jose F Velez. Offline continuous handwriting recognition using sequence to sequence neural networks. *Neurocomputing*, 289:119–128, 2018.

[46] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[47] Jianfeng Wang and Xiaolin Hu. Gated recurrent convolution neural network for ocr. In *NIPS*, 2017.

[48] Curtis Wigington, Seth Stewart, Brian Davis, Bill Barrett, Brian Price, and Scott Cohen. Data augmentation for recognition of handwritten words and lines using a cnn-lstm network. In *ICDAR*, 2017.

[49] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. 2018.

[50] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[51] Zhao Zhong, Junjie Yan, Wei Wu, Jing Shao, and Liu Cheng-Lin. Practical block-wise neural network architecture generation. In *CVPR*, 2018.