Lexicon and Attention Based Handwritten Text Recognition System

Lalita Kumari[1], Sukhdeep Singh[2], Vaibhav Varish Singh Rathore[3] and Anuj Sharma[1]
[1]*Department of Computer Science and Applications, Panjab University, Chandigarh, India*
https://anuj-sharma.in
[2]*D.M. College (Affil. to Panjab University), Moga, Punjab, India*
https://sites.google.com/site/sransingh13/
[3]*Computer Networking and Information Technology, PRL, Ahmedabad, Gujarat, India*
lalita@pu.ac.in, sukha13@ymail.com, vaibhav@prl.res.in, anujs@pu.ac.in

Abstract. The handwritten text recognition problem is widely studied by the researchers of computer vision community due to its scope of improvement and applicability to daily lives. It is a sub-domain of pattern recognition. Due to advancement of computational power of computers since last few decades neural networks based systems heavily contributed towards providing the state-of-the-art handwritten text recognizers. In the same direction, we have taken two state-of-the art neural networks systems and merged the attention mechanism with it. The attention technique has been widely used in the domain of neural machine translations and automatic speech recognition and now is being implemented in text recognition domain. In this study, we are able to achieve 4.15% character error rate and 9.72% word error rate on IAM dataset, 7.07% character error rate and 16.14% word error rate on GW dataset after merging the attention and word beam search decoder with existing Flor et al. architecture. To analyse further, we have also used system similar to Shi et al. neural network system with greedy decoder and observed 23.27% improvement in character error rate from the base model.

Key words: handwriting recognition, deep learning, word beam search, attention, neural network, lexicon.

1. Introduction

One of the major modes of communication is through handwritten text. From the 19^{th} century onwards, there is a rapid growth in the various computer technologies. One such domain is handwriting recognition [53]. It is a sub-domain of pattern recognition. The act of transcribing handwritten text into a digitized form is named as handwritten text recognition (HTR). This problem is widely studied in the research community due to its omnipresent need where people communicate and interact. The communication mode can be verbal, by sign language or with handwritten text. At present, handwritten text can be represented in two ways: online and offline. The online handwriting recognition is performed while the text to be recognized is written (e.g. by a pressure setting device), therefore temporal and geometric information is available. It is a way of processing and recognizing the text while writing or entering. The offline handwriting recognition is performed by collecting handwritten data and feeding these data into a computing device for recognition. The HTR task faces various challenges, like cursiveness of handwritten

text, different size and shape of each character and large vocabularies. There are many challenging aspects of any unconstrained handwriting recognition task. For example, in unconstrained handwriting recognition there is no control over author, writing styles and instrument used for writing. Moreover, different vertical and horizontal space present among different lines or different words in the same line causes uncertainty in a total number of lines and words, respectively, in a page and line of a handwritten document for an automatic text recognizer [33].

One of the key application areas of HTR is converting the ancient historical handwritten text into the digital form as a part of modern digital library. It helps in bridging the gap among computers and humans in various domains. Apart from this, invoices, notes, accidental claims, feedback forms are usually in the handwritten format and these can be used as digital footprints in the industrial domain. Despite from having developed many state of the art techniques, HTR domain is still far away from having a generic system that is able to read any handwritten text. Initial state of the art approaches of the HTR domain were based upon Hidden Markov Models (HMMs) [4, 9, 22, 24]. Recently, Neural Network (NN) based techniques have been on the rise among the state of the art methods, specially Convolutional Neural Network (CNN) techniques, in various challenging tasks.

In any typical HTR system an input image consists of a sequence of objects (characters) to be recognized, that contextually depend upon each other. The recognized text length also varies greatly; for example, English language word "To" is of length 2, "Tomorrow" is of length 8 and "The quick brown fox jumps over a lazy dog" is of length 33. The recognition of such sequences of objects is done either at word or line level. Although word level recognition systems are quite popular, for a generic system this technique is suboptimal. Firstly, for handwritten text, it is not always feasible to do word segmentation due to their proximity or partially overlapping locations. In addition to this, for densely written handwritten text, it is complex to detect large number of words and at last there may be scenarios where a word is not separated by a space which is a word separator considered by most of the linguistic systems [36]. Thus this study focuses on line level recognition architecture to make the system as generic as possible.

The CNN in connection with the RNN have been consistently performing good and are able to give state of the art results for the HTR problem [35, 38, 51]. The CNN typically has a convolutional layer that uses convolution operation for extracting the features of the given image by applying various filters. The RNN is used to capture sequences and contextual information hence is able to correlate the relationship among characters rather then treating them independently. The Bidirectional Long Short-Term Memory (BLSTM) [23, 27] RNN is used to capture long term dependencies and to consider more context in comparison to typical RNN architecture. Deep BLSTM RNN is widely used in speech recognition tasks [26, 46]. The output from BLSTM RNN is the character probability versus time matrix, where probabilities of specific characters are calculated

for each time step. In 2006, a technique was introduced to train and score NN architectures in which input sequence and output labels are given in the form of the input and output pair. The Connectionist Temporal Classification (CTC) function introduced in this technique does not depend upon the underlying network architecture [25]. Hence, in architecture trained by the CTC, output of the RNN is characters probabilities, including the blank character. A blank character is a special character introduced in CTC to handle duplicate characters. Obtaining the actual text from the output of the RNN is called decoding.

A typical HTR system includes various preprocessing steps to minimize the variation of text as much as possible. There are no generic preprocessing steps but these usually relie on the input of the HTR system [19].

In this study, we have usued the line based gated convolution text recognition system [15]. The attention was first introduced in Neural Machine Translation (NMT) task [2, 39]. In a typical NMT task attention is used to provide the importance of each word at a given time step while translating. Other than NMT, attention is also used in speech recognition tasks [14]. Some recent studies have also shown attention to be a promising method used with a HTR task. In this study, we have blended the attention mechanism with CTC based NN system to learn and propagate the image features and utilize the benefits obtained from both the techniques The key contributions of the present study are as follows.

- The text recognition system is explained in an end-to-end manner and in a simplified way.
- Popular attention mechanism is merged with existing state-of-the-art HTR techniques.
- Two separate NN systems are taken and merged with attention module to make our observation generalized.
- Attention module is merged with Flor et al. and Shi et al. architecture to learn and propagate the image features efficiently while training of the NN system.
- Additionally, Word Beam Search (WBS) decoder [49] has been added as a post processing step to improve the accuracy in Flor et al. system [15].
- Greedy decoding method is used with Shi et al. architecture [51] to show the percentage of improvement we observed by adding the attention module to the existing NN system.

The rest of the paper is organized as follows. Section 2 covers the key contributions in the domain of the HTR. In section 3 the system design is discussed. Section 4 briefs the experimental setup. In Section 5 the results of this study and the comparison with other works are presented. Section 6 contains the discussion. At last, conclusions are presented in Section 7.

2. Related Work

In this section, we have discussed the key contributions in the domain of the HTR. We especially focused on the methods and techniques involving line level HTR. Early

works in this domain have used dynamic programming based methods to segment and identify the words at character level using optimum path finding algorithm [3, 11]. Further improvement was observed by using the HMM based techniques in the HTR [55]. Standard HMM based methods lack handling the long sequences of characters as per the Markov assumption. To improve the accuracy further, these models are combined with other basic NN systems. In one such method, an HMM/ANN based architecture is presented in which trigram Language Model (LM) is used for recognition purposes [18]. Later, more advanced NN layers systems are studied in connection with the HMMs. In a similar study, the HMM based HTR architecture is used that improves the accuracy of recognition on IAM and RIMES dataset by applying preprocessing steps, discriminative HMM training and discriminative feature extraction. In [34], an LSTM network is used for feature extraction, and word and character level LMs are used to improve the accuracy even more. Further, in one such study, the activation function of the gated units of the LSTM is modified to make the overall system robust. A combination of HMM and LSTM is used as a recognizer [17].

Current state-of-the-art are NN based systems. In one such system, the Convolutional Recurrent Neural Network (CRNN) architecture is introduced that is a combination of the CNN and the RNN and is able to produce state-of-the-art results in recognizing sequential objects in scene-text images [51] (available also in [50]). Similar to CRNN, one variant of RNNs, that is Multidimensional Long Short-Term Memory (MDLSTM) network, has been widely studied by the research community [28]. The MDLSTM architectures provide the recurrence in both directions (horizontal and vertical) along the given image [37]. Thus, two dimensional data of unconstrained handwritten text can be processed. Utilizing this nature of the MDLSTM, many page level recognition systems of the HTR are also proposed. In one such study, the proposed architecture is able to recognize paragraph level texts. External segmentation of the paragraph into lines is prone to errors and these errors propagate from segmentation to recognition. By doing an implicit recognition, this study resolves the error generated due to poor segmentation. In this study, the MDLSTM-RNN is used with attention mechanism. In [6] (published earlier as [5]), a trigram LM is used that was trained on LOB, Brown and Wellington corpora [30]. In a similar work [8] (available also in [7]), authors used an attention based model for end-to-end HTR. In this approach, an MDLSTM, CTC and attention based model is gradually trained from consecutive images of words to a complete text line. As the size of the sentence increases accuracy increases gradually. To handle the paragraph data, augmentation techniques are used to generate enough training data and the model is modified with curriculum learning to adapt to recognition at paragraph level. These networks are complex and require the use of a large number of computational resources. The NN architecture based on convolutional and 1D-LSTM layers is able to learn similar features with a significantly smaller computational cost [45]. Some notable

state-of-the-art systems are only made up of CNN layers or attention techniques without any recurrent layer [13, 42, 43, 44, 52, 57, 58].

3. System Design

In this section, we presented an overview of each module proposed in the text recognition system in detail. In this study, we have used modified Bahdanau attention [2], which was successfully used in two state-of-the-art NN systems [15] and [51]. Both of these systems take the input image and extract its features. The propagated features are processed by the attention and the RNN layer to produce the occurrence probability of each character at each time step. Figure 1 presents the system architecture in detail. Later in the discussion section, more is written on the position of attention layer in the NN model.

3.1. Feature Extraction

In both above mentioned systems, a greyscale image is taken as an input and its feature map is produced as an output. A set of convolution layers is used to extract these input image features. The heart of a convolution layer is a convolution operation. A kernel slides over a given input image to produce its feature map. The convolution operation is followed by Rectified Linear Unit (ReLU) activation function, which produces the non-linearity in the NN system. Quick convergence was observed in ReLU comparison to the other activation function of the same class. As shown in Fig. 1 the input image is first pre-processed then this processed image is passed through a set of convolution and fully gated convolution layers in Flor et al. [15] to extract the most relevant features. While in Shi et al. [51], a series of convolution layers with varying kernel sizes are used to extract features of images.

3.2. Attention Module

The CNN output is a four dimensional vector which is reshaped to three dimensions (batch size, time-steps, features at each time step). These feature maps $(f_1, f_2..., f_n)$ are fed to the attention module as input. The motivation for applying attention is towards getting a more powerful representation using a weighted context vector (C). At a given timestep, it is computed using Eq. (1),

$$C = \sum_{i=1}^{T} s_i f_i^{\mathrm{CNN}}, \tag{1}$$

where $f_{\mathrm{i}}^{\mathrm{CNN}}$ is feature information at i_{th} time step, T is the total number of time steps and s_i is attention weight corresponding to $f_{\mathrm{i}}^{\mathrm{CNN}}$ which is calculated using Eq. (2),

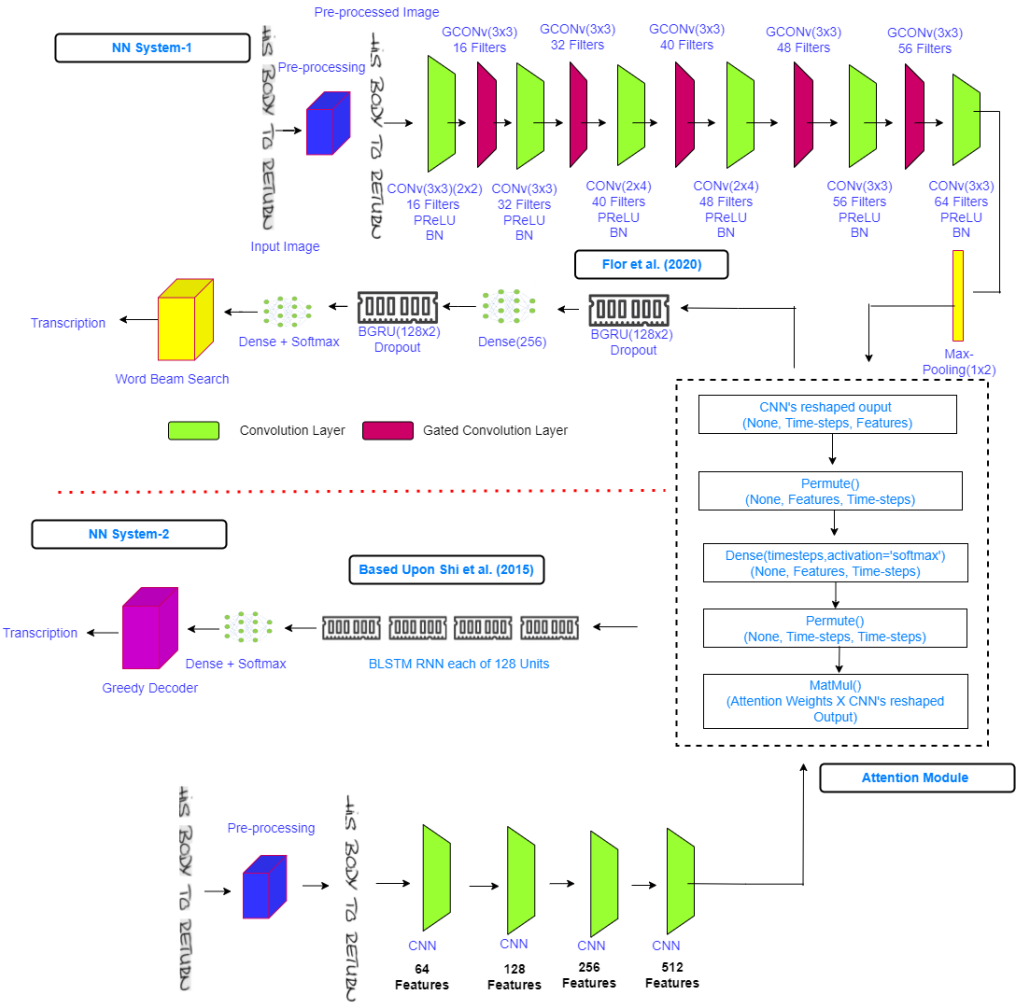

Fig. 1. System Architecture

$$s_i = \frac{\exp(a_i)}{\sum_{k=1}^{T} \exp(a_k)}, \tag{2}$$

where a_i is alignment score of feature f_i at a given time step. It is learned using Feed Forward Neural Network (FNN) while training of the NN model. The inputs of different

times are aggregated using attention based weighting. Thus, attention mechanism is applied in the time step dimension such that at each step more relevant features will be send to further RNN layers. Before applying the attention mechanism, we have to first permute the reshaped output of CNN. The current output without permutation will be of form (batch size, time steps, features per time step) and by applying dense layer without permuting it would be understood that there was no feature exchange due to attention mechanism, which is a false assumption. So, after the permutation operation, FFN layer is used to get attention probabilities. This operation will be performed for each time step. Thus we obtained a matrix that contains the weighted attention probabilities at each time step, which further multiplied to feature vectors to obtain the context vector. This context vector will be given to the next layer for further processing.

3.3. Recurrent Layers

Due to its property of having internal memory, the recurrent layers are widely used in sequence learning tasks. HTR is a sequence learning problem when we identify the probability of character occurrence given as an input image at each time step. As shown in Fig. 1, the output of the attention layer is given to Bidirectional Gated Recurrent Unit (BGRU) in Flor et al. system and similarly, the attention output is given to stacked BLSTM layers in Shi et al. system. Both systems used bidirectional stacked layers for processing input in forward and backward directions.

3.4. WBS Decoder

In a NN system, when trained with CTC loss function, the recurrent layer produces the character probabilities at each time-step. These probabilities are mapped to final character sequences using various decoding algorithms such as greedy, token passing and WBS decoder. This was proposed in [49]. The prefix tree made from the available corpus is internally used by this technique to decide which path to take while decoding at each time step. A *beam* is simply one possible character sequence. The *number* of beams that takes part in the next time step is equal to beam width except at $t = 0$. At the final time step, the beam that has the highest probability is selected as the final sequence and given as the output of the recognizer. We have used `Beam Width` $= 50$, and `Processing Mode` as `'NGram'` while applying this decoder in the present study [49].

4. Experimental Setup

In this section, we discuss the experimental work done in this study. This includes discussion on the datasets, preprocessing, data augmentation, evaluation metrics and training details. We have used the basic model of Flor et al. [15,16] and WBS according to [48,49]. The NN systems were implemented using the Keras package [29] in Python.

Tab. 1. Train, validation and test splits of IAM and GW datsets

SNo.	Dataset	Train set	Validation set	Test set
1	IAM Dataset (No. of characters=79)	6,161	900	1,861
2	GW Dataset (No. of characters=82)	325	168	163

4.1. Dataset

In this study, we evaluated our model on IAM [40] and GW [21] benchmarked datasets. These datasets contain pages of handwritten texts. These datasets contain the text images and their transcription at word, line and paragraph levels. Each of both datasets is discussed below.

4.1.1. IAM

The IAM dataset [40] contains the handwritten English text forms used by text recognizers for training and testing purposes. Data are present at word, line and paragraph levels. In this work, we have used the data present at line level with standard split as defined in Table 1. The IAM dataset contains 657 writers, 1539 pages of scanned text, 13353 text lines and 115320 words. All the data is in labelled format. LOB corpus is used to build the IAM dataset.

4.2. George Washington dataset (GW)

This dataset [21] contains the English letters written by George Washington to their associates in 1755. It has a total of 20 pages whose data is annotated at the word level, making 5000 words in total. We have used the train, test and validation split as specified in Table 1.

4.3. Preprocessing

The preprocessing techniques are used to improve the quality of degraded handwritten text documents. In this study, we have used illumination compensation [10], binarization [47] and deslanting [56] as preprocessing techniques. Figure 2 shows the image of the IAM dataset after each preprocessing technique.

4.4. Evaluation Metric

The evaluation metric is used to identify how well the proposed system is performing in comparison to earlier studies. We have used the standard evaluation metric Character Error Rate (CER) and Word Error Rate (WER). It is based on the Levenshtein

distance (LD). It is formulated as follows,

$$\text{WER} = \frac{S_{\text{word}} + D_{\text{word}} + I_{\text{word}}}{N_{\text{word}}} \tag{3}$$

where S_{word} is the number of substitutions required, D_{word} is the number of deletions required, I_{word} is the number of insertions required at word level, and N_{word} is the total number of characters in the ground truth sentence.

CER is the same as WER; the only difference is that in CER we work on the character level instead of the word level like in WER.

4.5. Data Augmentation

Data augmentation techniques are used to provide different variations of the samples available for training. For a NN system to be able to learn properly, the right amount of training data is required. The model can either be overfit or underfit based on the availability of the data. We have used random morphological and displacement transformations such as resizing, rotation, image displacement, erosion and dilation.

4.6. Training Details

In this section, we discuss the training and testing algorithm used in the present system. Algorithm 4.1 presents the training strategy and explains the details of the attention module. Algorithm 4.2 explains the decoding using the WBS decoder. In the Shi et al. system, the training architecture is the same except for the number and type of layers, but for the decoding purpose, we have used best path decoding to produce results in that setup.

Explanation We will discuss the training process as in algorithm 4.1 in line by line manner, as follows.

Fig. 2. Results of pre-processing techniques

Algorithm 4.1 Training Details

Input line images $I_1, I_2, ...I_n$ and ground truth $y_1, y_2, ...y_n$
Result Trained model weights on minimizing the validation loss

1: epochs=1000, batch=16, lr=0.001, stop_tolerance=20, reduce_tolerance=15; //initialize the training parameters
2: **procedure** ATTENTION(RNN_{out})
3: RNN_{out}=Permute(2,1)RNN_{out}; //permute the time and feature axis
4: α_p=Dense(timestep,softmax); //Calculate the attention weights
5: α_p=Permute(2,1)α_p;
6: $\text{Context}_{\text{vec}}$=Multiply($\text{RNN}_{\text{out}}$,$\alpha_p$); //Calculate the context vector
7: **return** $\text{Context}_{\text{vec}}$;
8: **end procedure**
9: **procedure** MAIN()
10: init model(); //Initialize the model framework
11: **for** i=1 to batch **do**
12: augmentImage(I_i); //Augment text line images
13: CNN_i=(I_i); //Extract features of the Image
14: Reshape_i=Reshape(CNN_i); //Reshape output of CNN for further proc.
15: $\text{RNN}_{\text{inp}-i}$=Attention($Reshape_i$); //Attention module
16: $\hat{y_i}$=RNN($\text{RNN}_{\text{inp}-i}$); //Processing RNN Layers
17: $\hat{y_i}$=Dense(timestep,$Num_{\text{char}} + 1$(for CTC blank)); //Finding the character occurrence at each time step
18: δ_{ctc}+=L_{ctc}(y_i,$\hat{y_i}$); //Computing CTC loss
19: **end for**
20: Backward(δ_{ctc}); //Updating model weights using backpropagation
21: **end procedure**

Line 1: Define training model parameters such as batch size learning rate, early stopping criteria and the total number of epochs.

Line 3: Permute the dimension of the output to enable feature exchange.

Line 4: Apply the dense() layer along the time step.

Line 5: Permute back the attention vector

Line 6: Obtaining the context vector by multiplying the features with the attention weights for each step.

Line 7: Returns the context vector to the main function to be further processed by RNN layers.

Line 10: Load the NN model.

Line 12: For a given batch, augment the preprocessed image.

Line 13: Extract the features of the image using a series of convolutions and gated convolutions operations.
Line 14: Converting 4D feature maps to 3D vectors to be further processed by attention and RNN layers.
Line 15: Applied the attention as defined above and obtained the context vector.
Line 16: Find the predicted character occurrence from the output of RNN
Line 17: Map the output of RNN to the number of characters of dataset + 1 for the sequence prediction.
Line 18: Compute the CTC loss from predicted and actual character sequence.
Line 20: Based upon the Loss value, train the NN system using backpropagation.

Algorithm 4.2 Prediction process

Input Text line image I, $E_{\text{test}_{\text{corpus}}}$, E_{chars}, $E_{\text{wordchars}}$
Result Prediction of image text with CER and WER

1: BW=50, mode='NGrams', smooth=0.01; //Initializing the WBS decoding params.
2: initNNModel(); //Loading of the trained model with all the layers
3: output=Modelpredict(I); //Predicting the text in the Image
4: $\hat{y}$= Decoder(BW,mode,smooth = 0.01, $D_{\text{test}_{\text{corpus}}}$, D_{chars}, $D_{\text{wordchars}}$); //Applying WBS decoding algorithm
5: CER,WER = accuracy(y,$\hat{y}$); //compute CER and WER

Explanation We will discuss prediction process as in algorithm 4.2 in line by line manner, as follows,
Line 1: Input text line image I. First, initialize the parameters of WBS decoding.
Line 2: Build the model.
Line 3: Process the image as per the trained model.
Line 4: RNN's output dimensions are swapped as per predefined input accepted by the WBS decoder.
Line 5: Computation of character occurrence using WBS decoding algorithm.
Line 6: Estimate the accuracy of the model on test images.

5. Results and Comparison

In this section, the results obtained in the present study have been discussed and compared with the other state-of-the-art methods. This HTR system recognizes the handwritten text on the line level, so the results are compared with other state-of-the-art line level systems. We are able to achieve 4.12% CER and 9.72% WER on the IAM dataset, and 7.07% CER and 16.14% WER on the GW dataset having Flor et al. as our based model and WBS as a decoding algorithm. We have also implemented Shi

et al. architecture and merged the attention module with it. Similar improvements were observed in that architecture reported in Table 4. The greedy decoder was used in this system. The given attention module is providing a 23.27% improvement with respect to the basic model.

Tab. 2. Comparison of present work with other state-of-the-art line level works on IAM Dataset

S No.	Reference	Method	CER	WER
1	Puigcerver et al. [45]	CNN + LSTM + CTC	4.4	12.2
2	Chowdhury et al. [12]	CNN + BLSTM + LSTM	8.1	16.7
3	Michael et al. [41]	CNN + LSTM + Attention	4.87	-
4	Kang et al. [31, 32]	Transformer	4.67	15.45
5	Yousef et al. [58]	CNN + CTC	4.9	-
6	Flor et al. [15]	CNN + BGRU + CTC	3.72	11.18
7	Present Work	CNN + Attention + BGRU + CTC	**4.15**	**9.72**

Tab. 3. Comparison of present work with other state-of-the-art line level works on GW Dataset

S No.	Reference	Method	CER	WER
1	Toledo et al. [54]	CNN + BLSTM + CTC	7.32	-
2	Almazan et al. [1]	Word Embedding	17.40	-
3	Fischer et al. [20]	HMM + RNN	20	-
4	Present Work	CNN + Attention + BGRU + CTC	**7.07**	**16.14**

Tab. 4. Similar NN System to Shi et al. recreated and attention module applied CNN as in Flor et al. except using greedy decoder instead of WBS

Architecture	CER(in %)
Model Based Upon Shi et. al	11.00
Model Based Upon Shi et al + Attention	8.44

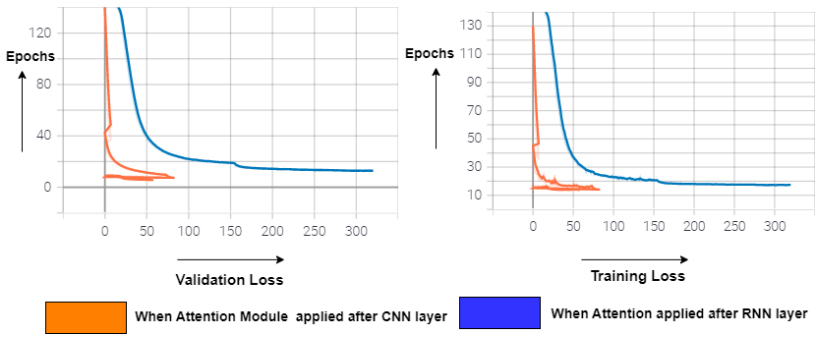

Fig. 3. Comparison of Training and Validation loss at different places of attention module

6. Discussion

In the proposed study, the attention mechanism helped in identifying relevant features in the given input image. We have experimentally found two possible positions where this attention block can be plugged in. Through extensive experiments and as per the graphs shown in Fig. 3, it is evident that the use of the attention mechanism before the recurrent layers and after the CNN layer helps the model to converge quicker with better accuracy. As shown in Fig. 3, with the same hyper-parameters for training, the model learns better and converges quickly when the attention module is applied after CNN layers.

7. Conclusion

In the present study, we have merged attention with two state-of-the-art NN systems that are Flor et al. and a small version of Shi et al. We were able to achieve 4.15% CER and 9.72% WER on the IAM dataset, and 7.07% CER and 16.14% WER on the GW dataset. We have also observed a 23.17% improvement in CER from the base model by applying attention module in the NN system similar to Shi et al. system. The accuracies obtained after applying the attention module favour our hypothesis that attention helps in learning the image features better. The position of applying the attention module is a critical step to consider, which we addressed in the discussion section. In future, we will work on extending this work to page or paragraph level.

Acknowledgements
This research is funded by the Government of India, University Grant Commission, under the Senior Research Fellowship scheme. The authors acknowledge PRL's supercomputing resource Vikram-100 (https://www.prl.res.in/prl-eng/hpc/vikram_hpc) made available for conducting the research reported in this paper.

References

[1] J. Almazan, A. Gordo, A. Fornes, and E. Valveny. Word spotting and recognition with embedded attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2552–2566, 2014. doi:10.1109/TPAMI.2014.2339814.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. 3rd Int. Conf. Learning Representations, ICLR 2015*, San Diego, CA, 7-9 May 2015. Accessible in arXiv. doi:10.48550/arXiv.1409.0473.

[3] R. E. Bellman and S. E. Dreyfus. *Applied Dynamic Programming*, volume 2050 of *Princeton Legacy Library*. Princeton University Press, 2015. doi:10.1515/9781400874651.

[4] A.-L. Bianne-Bernard, F. Menasri, Al-Hajj M. R., et al. Dynamic and contextual information in HMM modeling for handwritten word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):2066–2080, 2011. doi:10.1109/TPAMI.2011.22.

[5] T. Bluche. Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. *arXiv*, 2016. arXiv:1604.08352. doi:10.48550/arXiv.1604.08352.

[6] T. Bluche. Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. In *Advances in Neural Information Processing Systems 29 – Proc. 30th Conf. NIPS 2016*, volume 29, pages 838–846, Barcelona, Spain, 5-10 Dec 2019. Curran Associates, Inc. `https://proceedings.neurips.cc/paper/2016/file/2bb232c0b13c774965ef8558f0fbd615-Paper.pdf`.

[7] T. Bluche, J. Louradour, and R. Messina. Scan, Attend and Read: End-to-end handwritten paragraph recognition with MDLSTM attention. *arXiv*, 2016. arXiv:1604.03286. doi:10.48550/arXiv.1604.03286.

[8] T. Bluche, J. Louradour, and R. Messina. Scan, Attend and Read: End-to-end handwritten paragraph recognition with MDLSTM attention. In *Proc. 2017 14th IAPR Int. Conf. Document Analysis and Recognition (ICDAR)*, pages 1050–1055, Kyoto, Japan, 9-15 Nov 2017. IEEE. doi:10.1109/ICDAR.2017.174.

[9] T. Bluche, H. Ney, and C. Kermorvant. Tandem HMM with convolutional neural network for handwritten word recognition. In *Proc. 2013 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pages 2390–2394, Vancouver, Canada, 26-31 May 2013. IEEE. doi:10.1109/ICASSP.2013.6638083.

[10] K.-N. Chen, C.-H. Chen, and C.-C. Chang. Efficient illumination compensation techniques for text images. *Digital Signal Processing*, 22(5):726–733, 2012. doi:10.1016/j.dsp.2012.04.010.

[11] W.-T. Chen, P. Gader, and H. Shi. Lexicon-driven handwritten word recognition using optimal linear combinations of order statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):77–82, 1999. doi:10.1109/34.745738.

[12] A. Chowdhury and L. Vig. An efficient end-to-end neural model for handwritten text recognition, 2018. arXiv:1807.07965v2. doi:10.48550/arXiv.1807.07965.

[13] D. Coquenet, Y. Soullard, C. Chatelain, and T. Paquet. Have convolutions already made recurrence obsolete for unconstrained handwritten text recognition? In *Proc. 2019 Int. Conf. Document Analysis and Recognition Workshops (ICDARW)*, volume 5, pages 65–70, Sydney, NSW, Australia, 20-25 Sep 2019. doi:10.1109/ICDARW.2019.40083.

[14] A. Das, J. Li, G. Ye, et al. Advancing acoustic-to-word CTC model with attention and mixed-units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):1880–1892, 2019. doi:10.1109/TASLP.2019.2933325.

[15] A. F. de Sousa Neto, B. L. D. Bezerra, A. H. Toselli, and E. B. Lima. HTR-Flor: A deep learning system for offline handwritten text recognition. In *Proc. 2020 33rd SIBGRAPI Conference on*

Graphics, Patterns and Images (SIBGRAPI), pages 54–61, Porto de Galinhas, Brazil, 07-10 Nov 2020. doi:10.1109/SIBGRAPI51738.2020.00016.

[16] A. Flor de Sousa Neto. handwritten-text-recognition. GitHub repository, 2020. `https://github.com/arthurflor23/handwritten-text-recognition`.

[17] P. Doetsch, M. Kozielski, and H. Ney. Fast and robust training of recurrent neural networks for offline handwriting recognition. In *Proc. 2014 14th Int. Conf. Frontiers in Handwriting Recognition (ICFHR)*, pages 279–284, Hersonissos, Greece, 01-04 Sep 2014. IEEE. doi:10.1109/ICFHR.2014.54.

[18] P. Dreuw, P. Doetsch, C. Plahl, and H. Ney. Hierarchical hybrid MLP/HMM or rather MLP features for a discriminatively trained gaussian HMM: A comparison for offline handwriting recognition. In *2011 18th IEEE Int. Conf. Image Processing (ICIP)*, pages 3541–3544, Brussels, Belgium, 11-14 Sep 2011. IEEE. doi:10.1109/ICIP.2011.6116480.

[19] S. España-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez. Improving offline handwritten text recognition with hybrid HMM/ANN models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):767–779, 2010. doi:10.1109/TPAMI.2010.141.

[20] A. Fischer. *Handwriting Recognition in Historical Documents*. PhD thesis, Universität Bern, Switzerland, 13 Mar 2012. `https://www.researchgate.net/publication/259346163`.

[21] A. Fischer, A. Keller, V. Frinken, and H. Bunke. Lexicon-free handwritten word spotting using character HMMs. *Pattern Recognition Letters*, 33(7):934–942, 2012. Special Issue on Awards from ICPR 2010. doi:10.1016/j.patrec.2011.09.009.

[22] A. Fischer, K. Riesen, and H. Bunke. Graph similarity features for HMM-based handwriting recognition in historical documents. In *Proc. 2010 12th Int. Conf. Frontiers in Handwriting Recognition (ICFHR)*, pages 253–258, Kolkata, India, 16-18 Nov 2010. IEEE. doi:10.1109/ICFHR.2010.47.

[23] V. Frinken and S. Uchida. Deep BLSTM neural networks for unconstrained continuous handwritten text recognition. In *Proc. 2015 13th Int. Conf. Document Analysis and Recognition (ICDAR)*, pages 911–915, Tunis, Tunisia, 23-26 Aug 2015. IEEE. doi:10.1109/ICDAR.2015.7333894.

[24] A. Giménez, I. Khoury, J. Andrés-Ferrer, and A. Juan. Handwriting word recognition using windowed Bernoulli HMMs. *Pattern Recognition Letters*, 35:149–156, 01 2014. doi:10.1016/j.patrec.2012.09.002.

[25] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML '06: Proc. 23rd Int. Conf. Machine Learning*, pages 369–376, Pittsburgh, PA, USA, 25-29 Jun 2006. doi:10.1145/1143844.1143891.

[26] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proc. 31st Int. Conf. Machine Learning (ICML'14)*, volume 32 of *ACM Proceedings*, pages II–1764–II–1772, Beijing, China, 21-26 Jun 2014. JMLR.org. `https://dl.acm.org/doi/abs/10.5555/3044805.3045089`.

[27] A. Graves, M. Liwicki, S. Fernández, et al. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2009. doi:10.1109/TPAMI.2008.137.

[28] A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in Neural Information Processing Systems 21 – Proc. 22nd Conf. NeurIPS 2008*, volume 21, pages 545–552. Curran Associates, Inc., 2008. `https://proceedings.neurips.cc/paper/2008/file/66368270ffd51418ec58bd793f2d9b1b-Paper.pdf`.

[29] Keras Special Interest Group. Keras. simple. flexible. powerful. `https://keras.io`.

[30] S. Johansson, G. N. Leech, and H. Goodluck. *Manual of Information to accompany the Lancaster-Oslo/Bergen Corpus of British English, for use with digital Computers*. Department of English, University of Oslo, Oslo, Norway, 1978.

[31] L. Kang, P. Riba, M. Rusiñol, et al. Pay attention to what you read: Non-recurrent handwritten text-line recognition. *arXiv*, 2020. arXiv:2005.13044. doi:10.48550/arXiv.2005.13044.

[32] L. Kang, P. Riba, M. Rusiñol, et al. Pay attention to what you read: Non-recurrent handwritten text-line recognition. *Pattern Recognition*, 129:108766, 2022. doi:10.1016/j.patcog.2022.108766.

[33] G. Kim, V. Govindaraju, and S. N. Srihari. An architecture for handwritten text recognition systems. *International Journal on Document Analysis and Recognition*, 2(1):37–44, 1999. doi:10.1007/s100320050035.

[34] M. Kozielski, P. Doetsch, and H. Ney. Improvements in RWTH's system for off-line handwriting recognition. In *Proc. 2013 IAPR 12th Int. Conf. Document Analysis and Recognition (ICDAR)*, pages 935–939, Washington, DC, USA, 25-28 Aug 2013. IEEE. doi:10.1109/ICDAR.2013.190.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. doi:10.1145/3065386.

[36] L. Kumari and A. Sharma. A review of deep learning techniques in document image word spotting. *Archives of Computational Methods in Engineering*, 29(2):1085–1106. doi:10.1007/s11831-021-09605-7.

[37] L. Kumari, S. Singh, and A. Sharma. Page level input for handwritten text recognition in document images. In J. H. Kim et al., editors, *Proc. 7th Int. Conf. Harmony Search, Soft Computing and Applications (ICHSA)*, volume 140 of *Lecture Notes on Data Engineering and Communications Technologies*, pages 171–183, Seoul, South Korea, 23-24 Feb 2022. Springer Nature Singapore. doi:10.1007/978-981-19-2948-9_17.

[38] Y. Le Cun, B. Boser, J. S. Denker, et al. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems 2 – Proc. Conf. NeurIPS 2008*, volume 2, page 396–404, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc. `https://proceedings.neurips.cc/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf`.

[39] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proc. EMNLP 2015*, Lisbon, Portugal, 17-21 Sep 2015. Accessible in arXiv. doi:10.48550/ARXIV.1508.04025.

[40] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, 2002. doi:10.1007/s100320200071.

[41] J. Michael, R. Labahn, T. Grüning, and J. Zöllner. Evaluating sequence-to-sequence models for handwritten text recognition. In *Proc. 2019 IAPR Int. Conf. Document Analysis and Recognition (ICDAR)*, pages 1286–1293, Sydney, NSW, Australia, 20-25 Sep 2019. IEEE. doi:10.1109/ICDAR.2019.00208.

[42] J. Poulos and R. Valle. Character-based handwritten text transcription with attention networks. *Neural Computing and Applications*, 33(16):10563–10573, 2021. doi:10.1007/s00521-021-05813-1.

[43] A. Poznanski and L. Wolf. CNN-N-Gram for handwriting word recognition. In *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2305–2314, Las Vegas, NV, USA, 27-30 Jun 2016. doi:10.1109/CVPR.2016.253.

[44] R. Ptucha, F. Petroski Such, S. Pillai, et al. Intelligent character recognition using fully convolutional neural networks. *Pattern Recognition*, 88:604–613, 2019. doi:10.1016/j.patcog.2018.12.017.

[45] J. Puigcerver. Are multidimensional recurrent layers really necessary for handwritten text recognition? In *Proc. 2017 14th IAPR Int. Conf. Document Analysis and Recognition (ICDAR)*, pages 67–72, Kyoto, Japan, 9-15 Nov 2017. IEEE. doi:10.1109/ICDAR.2017.20.

[46] H. Sak, A. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Proc. Annual Conf. of the International Speech Communication Association (Interspeech)*, pages 338–342, Singapore, 14-18 Sep 2014. doi:10.21437/Interspeech.2014-80.

[47] J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000. doi:10.1016/S0031-3203(99)00055-2.

[48] H. Scheidl. CTCWordBeamSearch. GitHub repository, 2019. `https://github.com/githubharald/CTCWordBeamSearch`.

[49] H. Scheidl, S. Fiel, and R. Sablatnig. Word Beam Search: A connectionist temporal classification decoding algorithm. In *Proc. 2018 16th Int. Conf. Frontiers in Handwriting Recognition (ICFHR)*, pages 253–258, Niagara Falls, NY, USA, 5-8 Aug 2018. IEEE. doi:10.1109/ICFHR-2018.2018.00052.

[50] B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *arXiv*, 2015. arXiv:1507.05717. doi:10.48550/arXiv.1507.05717.

[51] B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2017. doi:10.1109/TPAMI.2016.2646371.

[52] F. Such Petroski, D. Peri, F. Brockler, et al. Fully convolutional networks for handwriting recognition. In *Proc. 2018 16th Int. Conf. Frontiers in Handwriting Recognition (ICFHR)*, pages 86–91, Niagara Falls, NY, USA, 5-8 Aug 2018. IEEE. doi:10.1109/ICFHR-2018.2018.00024.

[53] D. Suryani, P. Doetsch, and H. Ney. On the benefits of convolutional neural network combinations in offline handwriting recognition. In *Proc. 2016 15th Int. Conf. Frontiers in Handwriting Recognition (ICFHR)*, pages 193–198, Shenzhen, China, 23-26 Oct 2016. IEEE. doi:10.1109/ICFHR.2016.0046.

[54] J. I. Toledo, S. Dey, A. Fornes, and J. Llados. Handwriting recognition by attribute embedding and recurrent neural networks. In *Proc. 2017 14th IAPR Int. Conf. Document Analysis and Recognition (ICDAR)*, volume 01, pages 1038–1043, Kyoto, Japan, 9-15 Nov 2017. IEEE. doi:10.1109/ICDAR.2017.172.

[55] A. Vinciarelli. A survey on off-line cursive word recognition. *Pattern Recognition*, 35(7):1433–1446, 2002. doi:10.1016/S0031-3203(01)00129-7.

[56] A. Vinciarelli and J. Luettin. A new normalization technique for cursive handwritten words. *Pattern Recognition Letters*, 22(9):1043–1050, 2001. doi:10.1016/S0167-8655(01)00042-3.

[57] M. Yousef and T. Bishop. OrigamiNet: Weakly-supervised, segmentation-free, one-step, full page text recognition by learning to unfold. In *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 14698–14707, Seattle, WA, USA, 13-19 Jun 2020. IEEE. doi:10.1109/CVPR42600.2020.01472.

[58] M. Yousef, K. F. Hussain, and U. S. Mohammed. Accurate, data-efficient, unconstrained text recognition with convolutional neural networks. *Pattern Recognition*, 108:107482, 2020. doi:10.1016/j.patcog.2020.107482.

Lalita Kumari is pursuing her PhD in Computer Science with the Department of Computer Science and Applications at Panjab University Chandigarh under UGC Senior Research Fellowship Scheme. Her research interest includes Machine Learning and Pattern Recognition in the area of cursive handwriting recognition. She is also interested in other areas of Pattern Recognition, Machine Learning and Image Processing.
ORCID: 0000-0002-4406-3324 .

Sukhdeep Singh is a doctor in Computer Science and works at DM college Moga. He has done his postdoctoral research at Boise State University, USA and received PhD in Computer Science from DCSA Panjab University Chandigarh. He has also done his MCA from DCSA Panjab University. His research interest includes Pattern Recognition and Machine Learning, especially in handwritten text recognition.
Homepage: https://sites.google.com/site/sransingh13/ .

Vaibhav Varish Singh Rathore is working with PRL Ahmedabad. He graduated from the National Institute of Technology Allahabad, India. His research interests include Pattern Recognition, Machine Learning, High-Performance Computing, Networking and Cyber Security.
ORCID: 0000-0003-2045-5339 .
Homepage: https://www.prl.res.in/~vaibhav .

Anuj Sharma is working with DCSA Panjab University Chandigarh. He has done his Post-Doc at RWTH Aachen, Germany and received PhD in Computer Science from Thapar Institute of Engineering and Technology, India. His research interests include Pattern Recognition and Machine Learning in the areas of Document Analysis, Handwriting and Image Recognition.
Homepage: https://anuj-sharma.in .