



Refocus attention span networks for handwriting line recognition

Mohammed Hamdan¹ · Himanshu Chaudhary² · Ahmed Bali³ · Mohamed Cheriet¹

Received: 1 June 2022 / Revised: 24 July 2022 / Accepted: 9 December 2022 / Published online: 25 December 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

Recurrent neural networks have achieved outstanding recognition performance for handwriting identification despite the enormous variety observed across diverse handwriting structures and poor-quality scanned documents. We initially proposed a BiLSTM baseline model with a sequential architecture well-suited for modeling text lines due to its ability to learn probability distributions over character or word sequences. However, employing such recurrent paradigms prevents parallelization and suffers from vanishing gradients for long sequences during training. To alleviate these limitations, we propose four significant contributions to this work. First, we devised an end-to-end model composed of a split-attention CNN-backbone that serves as a feature extraction method and a self-attention Transformer encoder–decoder that serves as a transcriber method to recognize handwriting manuscripts. The multi-head self-attention layers in an encoder–decoder transformer-based enhance the model's ability to tackle handwriting recognition and learn the linguistic dependencies of character sequences. Second, we conduct various studies on transfer learning (TL) from large datasets to a small database, determining which model layers require fine-tuning. Third, we attained an efficient paradigm by combining different strategies of TL with data augmentation (DA). Finally, since the robustness of the proposed model is lexicon-free and can recognize sentences not presented in the training phase, the model is only trained on a few labeled examples with no extra cost of generating and training on synthetic datasets. We recorded comparable and outperformed Character and Word Error Rates CER/WER on four benchmark datasets to the most recent (SOTA) models.

Keywords Split attention convolutional network · Multi-head attention transformer · Seq2Seq-model · BiLSTM · Line handwriting recognition

1 Introduction

Handwriting Text Recognition (HTR) systems allow computers to read and understand human handwriting. HTR is useful for digitizing the textual contents of old document images in historical records and contemporary administrative material such as cheques, law letters, forms, and other documents. While HTR research has been ongoing since the early 1960s [34], it remains a challenging and unsolved research problem. The fundamental problem is the wide range of variations and ambiguity encountered by different writers when crafting words. Because the words to be deciphered usually adhere to well-defined grammar rules, it is possible to eliminate gibberish hypotheses and enhance recognition accuracy by modeling the linguistic practices. HTR is usually embarked with a blend of computer vision and natural language processing (NLP).

In nature, handwritten text is a signal that follows a particular sequence. Texts in Latin languages are written in

✉ Mohammed Hamdan
mohammed.hamdan.1@ens.etsmtl.ca

Himanshu Chaudhary
him4318@gmail.com

Ahmed Bali
ahmed.bali.1@ens.etsmtl.ca

Mohamed Cheriet
mohamed.cheriet@etsmtl.ca

¹ Synchromedia Lab, System Engineering, University of Quebec (ETS), 1100 Notre-Dame St W, Montreal, Quebec H3C 1K3, Canada

² Data Science, Dr. A.P.J. Abdul Kalam Technical University, CDRI Rd, Naya Khera, Jankipuram, Lucknow, Uttar Pradesh 226031, India

³ Department of Software and IT Engineering, University of Quebec (ETS), 1100 Notre-Dame St W, Montreal, Quebec H3C 1K3, Canada

the left to the right direction, whereas non-Latin scripts are written from right to left; an ordered sequence of letters creates the words in both languages. As a result, HTR approaches used temporal pattern recognition techniques to overcome and maintain the sequence order. The usage of Deep Learning techniques progressed from early techniques based on Hidden Markov Models (HMM), with Recurrent Neural Network (RNN) and its variations Bidirectional Long Short-Term Memory (BiLSTM) networks becoming the mainstream alternative. Sequence-to-Sequence (Seq-Seq) techniques, served by encoder–decoder networks directed by attention mechanisms, have recently begun to be involved in HTR, inspired by their success in applications such as speech-to-text or automatic translation. It is not only feasible to decode images sequentially using the methods mentioned above, but it is also conceivable to learn which characters are more likely to follow each other in the decoding process. Language modeling can only increase recognition performance if applied as a post-processing step.

One crucial flaw persists despite the success of attention-based encoder–decoder architectures in HTR. These attention techniques involved in RNN's variations, either LSTMs or GRUs, are utilized on the standard Convolutional Neural Network (CNN) feature extraction method. Due to memory constraints, the sequential approach discourages parallelization during training and substantially reduces processing speed while processing more extensive sequences.

To alleviate those flaws, we were inspired by End-to-End Object Detection with Transformers (DETR) architecture [8]. Though the DETR was meant to detect objects, we exploited their architecture to solve the HTR task. We explore the impact of slant removal and illumination computation preprocessing techniques. Also, we were inspired by the findings as mentioned earlier, [54] innovative work on split attention on convolutional layers (ResNeSt) and Transformer encoder–decoder architectures, respectively. Transformers and ResNeSt are based on attention mechanics, with no recurring techniques. Motivated by this benefit, we propose addressing the HTR problem with an architecture involving attention mechanisms on feature extractions and encoding–decoding feature representations. We desire to address both the suiTable phase of character recognition from images and understand the corresponding language model dependencies of the character sequences. The encoder–decoder Transformer multi-head self-attention module decoded textual ground truth at both stages of the visual feature representations.

Transformers have demonstrated superior performance to recurrent networks in various language and visual applications while more parallelizable than GRUs and BiLSTMs, requiring less training time. In contrast to classic speech and translation recognition models, our proposed split attention transformer networks operate at the character level rather

than the word level. Therefore, no specified fixed vocabulary is required with our architecture. Thus, the proposed model can recognize words that our training model has never seen, known as out-of-vocabulary (OOV) terms. We obtained competitive CER and WER performance compared to the current existing methods of the state-of-the-art (SOTA) outcomes on four publicly available English script (IAM, Bentham, and Washington) and French script (RIMES) datasets with no synthetic training data. Our contributions to this paper are:

- We experimented with different backbone feature extraction methods; Table 4 shows the findings of the standard CNN-BiLSTM model and Self Attention module-based CNN-BiLSTM where the attention mechanism improved the performance. Whereas the impact of the deeper layer networks between ResNet50 and ResNet101 coupled with self-attention BiLSTM as shown in Table 6 the deeper the layers, the better the performance. From there, we examine the standard CNN feature extraction and the attention-based CNN feature extraction methods as shown in Table 7. As a result, the split attention Transformers have demonstrated superior performance to recurrent networks in various language and visual applications while being more parallelizable than GRUs and BiLSTMs, requiring less training time. In contrast to classic speech and translation recognition models, our proposed split attention transformer networks operate at the character level rather than the word level. Therefore, no specified fixed vocabulary is required with our architecture. Thus, the proposed model can recognize out-of-vocabulary words (OVV) that have never been seen during the training. Our experimental results demonstrate a comparative performance accuracy of CER and WER compared to the state-of-the-art (SOTA) outcomes on the publicly available IAM, RIMES, Bentham, and Washington datasets with no synthetic training data. Since the convolution network ResNeSt101 outperforms the standard ResNet101 convolution network. Thus, we choose the ResNeSt101 for feature extraction method on the proposed model.
- We investigated the influence of different preprocessing approaches such as illumination adjustment, removing cursive handwriting, and combining both. Table 5 shows that only removing the cursive writing improves the performance while using the illumination compensation technique declines the performance of the raw dataset. Thus, we opt only for the recursive text technique to remove slanted and sloped text from our datasets before feeding them to the CNN for feature extractions.
- To the best of our knowledge, the proposed approach is considered the first study to examine the implications of attention mechanisms on both ResNeSt split attention convolution feature extraction–Transformers multi-head

attention encoder–decoder for the HTR problem without using any recurrent architecture. Specifically, we want to extract, encode, and decode robust representations from document images and model language using a unified architecture that can detect character sequences while providing context to differentiate between letters or words that may appear similar. The proposed architecture operates on a character-by-character basis, avoiding the need for predetermined lexicons, known as a lexicon-free model.

- Using pre-trained weights from ImageNet as a starting point benefits our model's rapid convergence and learning. Pre-training with the Bentham dataset also allows for competitive outcomes with a minimum amount of annotated training data.
- On the benchmark IAM, RIMES, Washington, and Bentham datasets, our proposed HTR model improves SOTA performance with few label data. In Sect. 5.2, we conduct thorough ablation and comparison investigations to demonstrate the efficiency of our model. Finally, we showcase the influence of using another popular pre-trained model (the CLIP) on the IAM dataset.

The remaining of this paper is organized as follows. Section 2 presents the most recent methods related to the one presented here regarding the tackled problem and modeling choices. In Sect. 3, the proposed method gives the system's details. Experiments are reported in Sect. 4, followed in Sect. 5 presents a discussion of our findings and compare to the state of the art, in sect. 6 we conclude how the system could be improved and present the challenge of generalizing it to complete documents.

2 Related work

The sequential pattern of the recognition framework has traditionally been used to recognize handwritten text. Text-line images are handled by learning models using their internal states to analyze incoming signals in variable-length lines. The HTR applications follow the same process paradigm using either Hidden Markov Models (HMM) [5,21,41] or Deep Neural Networks architecture-based (DNN) such as Bidirectional Long Short Memory (BiLSTM), multidimensional LSTM (MDLSTM) and Gated Recurrent Units (GRU) accompanied with Connectionist Temporal Classification (CTC) which used to label unsegmented sequences of handwritten images with RNN-LSTM [23,24,43], encoder–decoder networks (seq2seq) [35], nonrecurrence transformer networks [30]. Recently, attention mechanisms [3,13,25,51] have emerged as an essential component of any model that must account for global dependencies. In respective, self-attention [15,38] calculates the response or the weight at a

particular sequence location by paying attention to the entire sequence of the contributions at that position. The machine translation model has demonstrated that cutting-edge results can be achieved using only self-attention in the literature.

2.1 Recurrent network: CTC encoder–decoder

Recently, the HTR task was treating as a sequence-to-sequence (Seq2Seq) model [1,35,47,55]. Seq2Seq model transforms a sequence of convolutional and recurrent text image segments into transcribed text. The training of networks that utilize this strategy can be accomplished in one of two ways: first to maximize the categorical cross-entropy loss, then to combine that loss with the recurrent CTC loss. [22] employed LSTM and CTC for text-line recognition with no prior word-level segmentation. LSTM and CTC techniques help improve recognition applications of handwriting documents segmentation-free. Many studies [24,29] proved that the LSTM model effectively predicts longer sequences than standard RNN. For instance, the cell gates in the LSTM architecture allow it to remember important information from inputs that have already passed through, which distinguishes it from the RNN. More improvement in recent years is well investigated in the BiLSTM model [16,46]. While the information from both forward and backward (backpropagation) is used as input to the final output layer, BiLSTM provides an additional training capability that improves prediction accuracy. Therefore, BiLSTM can detect and extract more time dependencies and resolve them more precisely. While crossing over the input image, fixed grids specify convolutional kernels and focus on all input pixels simultaneously, disregarding the challenges of handwritten text like inter/Intra class variations, scale, and orientation, and the importance of ink pixels. Authors [9] proposed a convolutional neural network with a deformable variation in its convolutions. This variation of CNN can deform based on its image input and better geometrically respond or adapt to the variations in the textual contents.

A Separable Multidimensional Long Short-Term Memory (SepMDLSTM) was applied by Chen et al. [14] to encode the input text line pictures for script identification and multiscript text recognition via convolutional feature extraction. Pham et al. [39] used Multidirectional LSTM layers with CTC for computing the negative Log-likelihood for sequences. Pham investigated how dropout can work with recurrent and convolutional layers in deep network architecture, mainly on word handwriting recognition. Furthermore, they examine the proposed method of line recognition with language modeling and lexicon constraints. Wigington et al. [49] used random perturbations on a regular grid as an augmentation technique and a novel profile normalization technique on word and line handwriting text. Those techniques help the performance of their proposed CNN-LSTM model. Bluche et al. [6] proposed

a generic model based on a convolutional encoder of the input images and a bidirectional LSTM decoder predicting character sequences. The authors also proposed a convolutional gate in the decoder in order to control the propagation of the next layer's feature representation. Aradillas et al. [2] have comprehensively investigated the various combination of (TL and DA) techniques on CNN for feature extraction accompanied with 2DLSTM layers for classification. Puigcerver et al. [43] reported an efficient and outperformance accuracy on text line recognition model based on convolutional and 1D-LSTM rather than 2D-LSTM which improve the speed of their proposed model. In our recurrent baseline model, we employed BiLSTM on top of different backbone feature extraction methods, including CNN with/without attention mechanism. Unlike the existing studies, we answered different hypothesis questions regarding the recurrence modelling performance and architecture.

2.2 Non-recurrent network: transformer encoder–decoder

Transformer [4] is one of the most successful models for processing long sequences. While transformers were mainly focused on Natural Language tasks like BERT [19] and GPT-3 [7], they are widely spread in the Computer Vision communities to tackle various tasks on the domain. Transformer Encoder–Decoder networks are composed of three main components: positional encoding, self-attention multi-head attention modules, and the feed-forward layers. The main advantage of Transformer over RNN is that the former process long sequences in parallel. [40] exploits the Transformer for action recognition model where the problem is close to the handwriting recognition task as both datasets suffer from inter-class and intra-class variations. The SOTA frameworks of handwritten text recognition using BiLSTM have achieved acceptable recognition results, but the training process is too computationally intensive. Furthermore, even though they are supposed to describe language-specific dependencies [19,20], they often fall short and require additional post-processing processes. For the first time, we propose avoiding any recurrent architecture by using split attention on the CNN and a multi-head transformer network for the HTR task. It is possible to detect long character sequences from images and model language at a character level, avoiding established lexicons.

Kang et al. [30] utilized a deep CNN as a feature extraction method and the transformers as a transcriber for handwriting recognition tasks. They tested their proposed method on the IAM dataset with different methods, including language models of the IAM and WikiText. Flor et al. [17] proposed a novel and efficient architecture for HTR constructed on Gated-CNN and integrated with two steps of the language model at the character and word levels. [9] authors used a

deformable convolutional-recurrent network in order to adapt geometric transformations rather than a standard CNN as a backbone for their HTR model inspired by CRNN and 1D-LSTM. Further to what we found in the state-of-the-art methods, we built on top of the most efficient architectures and transformed them to solve HTR problems efficiently and effectively in a straightforward approach.

2.3 Data augmentation and transfer learning

Research studies in deep learning [31,50] show the impact of DA by transformations I input image and preserving the original class for new perturbing images such as flipping, resizing, cropping, rotation, and scaling. DA creates different copies of images for unique ground truth (label). Authors of [42] used affine transformation to generate more to the input image, another 36 images. They were utilizing both rotation and shearing where rotation was used around the centered input image with these specific angles ($-5, -3, -1, +1, +3, +5$) while the shear on ($-0.5, -0.3, -0.1, +0.1, +0.3, +0.5$) degrees. Wiginton et al. [49] proved two DA techniques that help to reduce CER and WER for handwriting text recognition. They applied the profile normation technique to make neural networks more tolerant, just like a human being when reading text that has various variations. [11] proposed two DA methods on the training set: (1) crafted multiscaled data, which proved to boost the training performance for HTR with fewer labeled data. (2) normalization scheme model-based that addresses the problem of handwriting variability at the recognition phase. The DA techniques of these studies are applied to a relatively large known dataset. However, the regularization impact of the DA technique has little effect if only one writer of a small database is changed. Therefore, we must combine TL and DA to reduce the final error rate, especially for a small dataset.

3 Proposed method

This section presents the baseline convolution neural network model and the preprocessing methods used in our experiments. Then, we propose the split-attention convolutional-transformer architectures that we employ in the experiments.

3.1 Baseline recurrent model

We embrace the baseline model inspired by Puigcerver [43] where the primary purpose of this model is to validate our hypothesis on attention mechanism and the impact of preprocessing techniques. We hypothesized that using an attention mechanism in the convolutional layer help to retrieve more robust image-line representations than non-attention CNN.

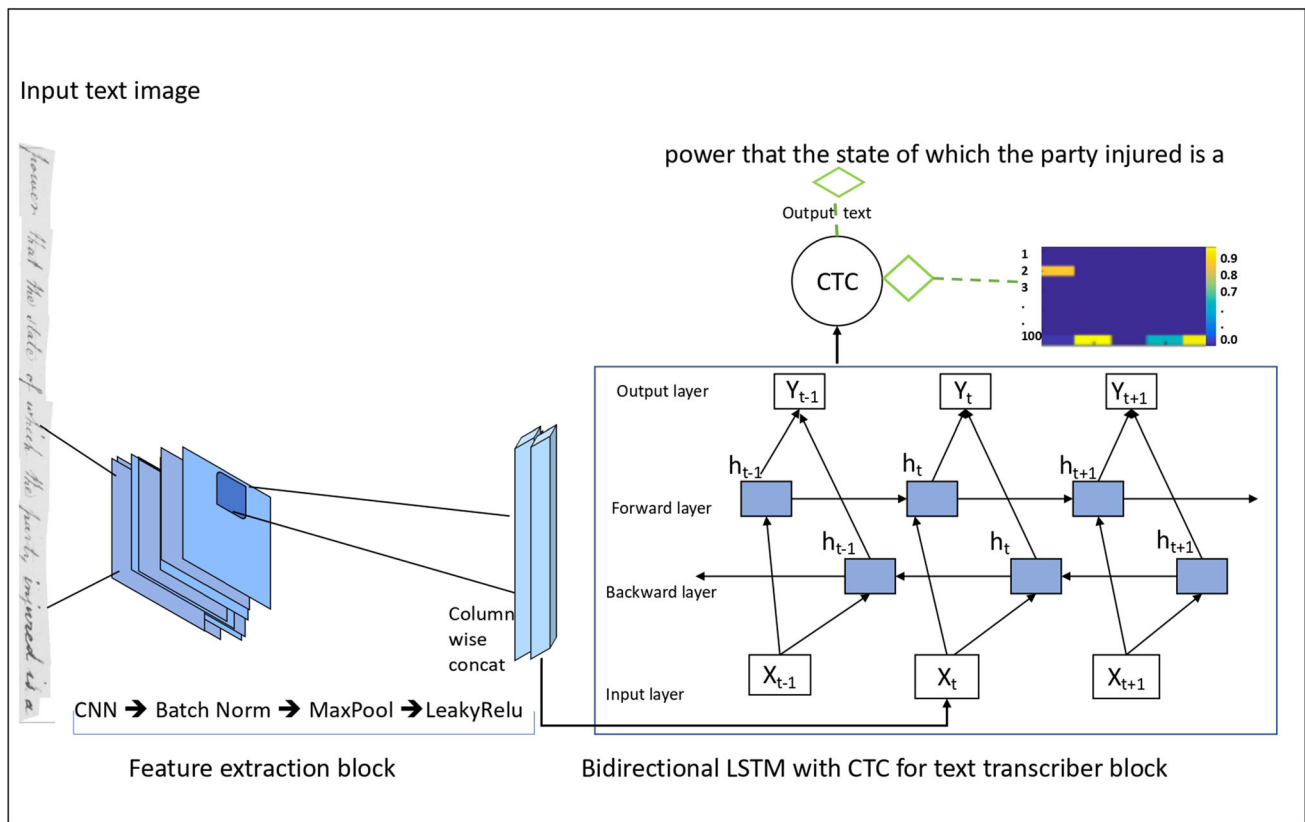


Fig. 1 An overview baseline model: an input line image is fed to five blocks of convolution layers with 3*3 filters each, batch normalization for stability and fast convergence, max-pooling for down-sampling, and

leakyReLU activation function for nonlinearity serve as feature extraction block then followed by five blocks of BiLSTM serves as text transcriber with Dense layer, Softmax, and CTC decoder

Figure 1 gives an overview of the baseline HTR model. The proposed based line model takes a text image as input. Then, there are four main parts to the model: the feature extracted from a given input image using the ConvNet method, a self-attention module to enhance the extracted feature representations, the visual representation treated as sequences and outputted their corresponding character probability distributions using BiLSTM method, finally, the last textual transcription obtained from the decoding block using CTC loss/decoding method. We train the baseline model by ensuring that the output sequence of CTC probability distribution is as high as possible. So, in addition to the characters that make up the text, the RNN gets a unique blank character that means “no other character.”

We have adopted the self-attention module from [53]. After experimenting with the baseline model, we found that adding a self-attention module to the end of CNN obtained the most robust feature representations and improved model performance. The baseline Handwriting recognition model is trained using CTC loss and further detailed in experimental setup Sect. 4.3.1. The CTC tackles the issue of alignment, as alignment information is not provided for the input data

or output transcription since handwriting differs from person to person. The influence of the attention mechanism on the CNN feature extraction method is displayed in Table 4. This method reduced 0.65% on the Bentham dataset while injecting the attention module into the proposed baseline model.

3.2 Preprocessing

The used datasets have a wide range of image sizes. In contrast, the input dimension of models must be fixed for efficient training and a pooling layer. Therefore, images in this paper are downsized to 1024 height and 128 widths with maintaining the aspect ratio. Since we have black and white text-line images, pixels are either 0 or 255. All of the characters and the background in the images are represented as 0 and 255, respectively. So we have added white color padding to the image in order to maintain the aspect ratio after resizing. More specifically, adding the padding where the pixels were short of the desired resolution; otherwise, we first resize along the shorter dimension such that the width is 128 pixels and then, crop the image along the height, such that the height is 1024 pixels. Maintaining a consistent aspect ratio allows our

Convolutional Neural Network to learn more discriminative and compatible features.

Various noises often affect offline handwritten text images scanned by various scanned devices. Also, different writers' handwriting styles (fonts) vary extremely widely. Though our proposed model is an end-to-end HTR system, input images are cleaned using the deslanting method [48] to remove the writing cursive style. Illumination Compensation technique [12] to remove shadows and balance brightness and contrast normalization before feeding to our CNN for feature extraction. In the present system, the following collection of preprocessing operations has been applied (i) illumination compensation techniques, (ii) removal of slant and slope from cursive handwriting text, (iii) normalization of line height to a fixed value (128 units), keeping the aspect ratio unchanged and (iv) a combination of both (i, ii).

The Illumination Compensation Techniques are five processes to balance uneven light distribution. The primary goal of the entire process is to produce content with a high degree of recognition. To eliminate the slanted and skewed text, we extend the most popular technique used as a new normalization approach to solving the cursive text in the word-level handwritten text by removing the slope and the slanted text. The results of those preprocessing experiments are conducted on the Bentham dataset and presented in Sect. 5.1.2.

3.3 Transformer-based model

In this subsection, we explained in detail the proposed HTR convolutional transformer architecture, starting with the feature extraction backbone and then, the transformer component methods.

3.3.1 Feature Extraction using Convolutional Neural Network

We investigate different CNN architectures, specifically the most recent SOTA models, including ResNet [27] and ResNeSt [54] with an attention mechanism for better feature extraction. Tables 6 and 7 report the results on the Bentham dataset as we found out that when the network goes deeper and uses the attention, the better the feature extraction is and hence, the better accuracy. Despite recent advancements in image classification models [31], most downstream applications such as image recognition [27], object detection [45], and semantic segmentation [32] continue to utilize ResNet variations as the backbone network due to their simplicity and modularity. In contrast, we employ a basic and modular Split-Attention block that allows attention to be distributed across feature-map groups. Stacking these Split-Attention blocks on ResNet-style formed a new variant called ResNeSt. ResNeSt is chosen because it preserves the overall ResNet structure in subsequent tasks without charging extra processing costs.

ResNeSt beats other SOTA neural networks with comparable model complexity and improves downstream HTR and OCR tasks.

3.3.2 Encoder–Decoder Transformer Network

In this subsection, we describe the main components of the proposed transcriber model architecture:

Encoding feature representation: High-level representations of features are extracted from the line handwritten image ($x_i \in X$) in CNN feature extraction sub-network 3.3.1, where x_i represents a sample of input images as in Fig. 2. The visual representation and sequential order information are encoded. To process the handwritten line images, the input image x_i is first processed by the CNN, which can handle images of any size. We can obtain an intermediate visual feature representation F_v of size ($f = 2048$) as a result of the process. The ResNeSt101 convolutional architecture serves as the backbone of our convolutional architecture. Visual feature representations using standard CNN cannot compact the global representation of the input image; therefore, the attention layer in ResNeSt101 helps extract this meaningful information.

Positional encoding (PE): In Latin scripts, text images are processed sequentially from left to right. While avoiding repetition, Positional Encoding stages before the transformer encoder phase are meant to leverage and encode such crucial information. It also helps the model define the exact location of the next word and character in the text line. For the model to utilize the sequence's order, we employ the positional encoding (PE) of tokens within the sequence. As a result, we incorporate the input embeddings with the PE at the bottom of the decoder, as shown in Fig. 2. The encoded feature vector from the ResNeSt is converted to the same hidden dimension (256) as the transformer encoder before adding the PE. The PE in Eq. (1, 2) are useful in the proposed architecture since they teach our model wherein the sequence is actively focused.

$$PE_{(\text{pos}, 2i)} = \sin\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \quad (1)$$

$$PE_{(\text{pos}, 2i+1)} = \cos\left(\frac{\text{pos}}{10000^{\frac{2i+1}{d_{\text{model}}}}}\right) \quad (2)$$

where pos represents the character at a particular position i , d_{model} is the size of the hidden dimension in the transformer block; both Eq. (1, 2) are used at the input of decoder to add the absolute positional information. We have used the 2D learnable positional encoding for the encoder to get the rich features for character representation.

Decoding visual feature representation: The self-attention is used to further improve the visual features of the

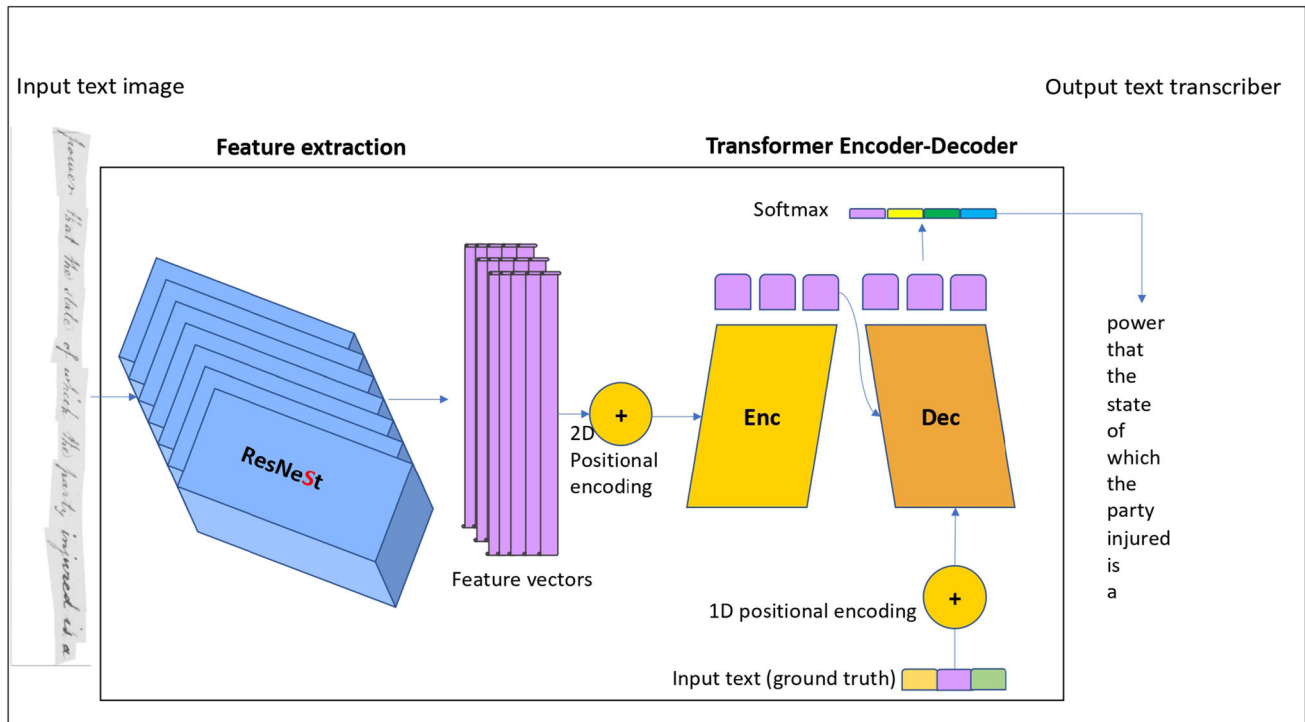


Fig. 2 Proposed model architecture: the Split attention of CNN serves as a feature extractor method where an input line image is fed to the ResNeSt101 backbone that outputs the feature vector, which will be coded using the position encoding technique before providing to the

transformer encoder part, and an encoder–decoder transformer network serve as text transcriber method where regularization, batch normalization, and Positional encoder are essential components of the architecture

extracted representations. This module accepts three inputs that are equivalent to \hat{F}_v that represented Q_v, K_v, V_v , for the query, the key, and the value, respectively. Let A_v^i be the correlation information of the attention visual feature obtained by Eq. 3.

$$A_v^i = \text{softmax} \left(\frac{q_v^i \cdot K_v}{\sqrt{f}} \right) V_v \quad (3)$$

where the $q_v^i \in Q_v$ as input query and i range from $(0$ and $w - 1)$, K_v and V_v are the input key and value, respectively. Finally, we obtain the high level visual representation from Eq. 3 where $\hat{F}_v = \{A_v^0, A_v^1, A_v^2, \dots, A_v^{w-1}\}$. In Fig. 3, we visualize the role of self-attention on the target line image. Due to the short length of that line sequence, we also show the visualization of the padding. The padding shows at the end of the presented image until we reach the maximum length of the input image on the datasets, which is pre-determined as 2048.

Transcriber text decoding : As depicted in Fig. 2 where visual aspects and language-specific information gathered through textual representations are taken care of this component. It outputs the decoded characters and anticipates the next likelihood character following decoded characters sequences. To analyze the line string effectively, we need

symbols that do not contain textual information and the numerous characters inspected in the vocabulary size V alphabet. The characters $< BOL >$ and $< EOL >$ are used to indicate the beginning and end of the line sequence, respectively. In contrast, the character $< pad >$ is used to indicate padding, as depicted in Fig. 3. Except for the initial/first character, the transcriptions ($y_i \in Y$) are padded out to a maximum predicted many characters of N . Character level embedding uses a dense layer to convert every character from the input string into an f_d -dimensional (f_d) vector. Eq. 4 uses the same positional encoding as in Eq (1, 2) where the PE should encode time-steps uniquely. In Fig. 3, we visualize the role of self-attention on the target line image.

$$F_d = \text{Mapping}(y_i) + PE \quad (4)$$

where y_i is the ground-truth text transcription, PE is the positional encoding.

4 Experimental setup

In this section, we examine whether the proposed method (ResNeSt101) is suitable to tackle the problem of the HTR

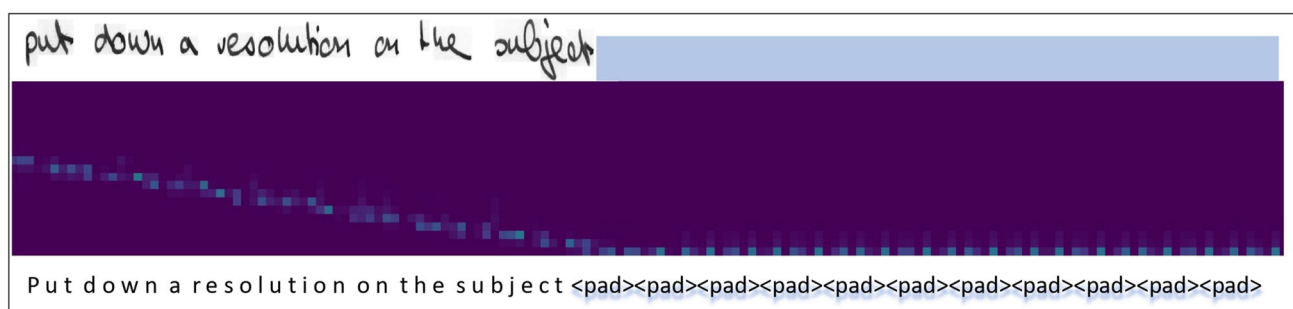


Fig. 3 The impact of self-attention in transformer-decoder network demonstrating the padding for this input line image

Table 1 Descriptions of the partitions on used datasets

Database	Training	Validation	Testing
IAM	6161	900	1,861
RIMES	10193	1,133	778
Washington	325	168	163
Bentham	9195	1415	860

task based on split attention convolutions compared to the other baseline models.

4.1 Datasets

This study examines the HTR system across four datasets: the IAM [33], RIMES [26], Washington [28], and Bentham [10] databases. Table 1 describes the standard partitions used by other researchers on the four datasets to ensure a fair comparison with the current work. Precisely, we can describe the used datasets as follows.

The IAM database has 13353 modern English text lines created by 657 distinct authors. The RIMES database is a French handwriting script collected by 1,300 writers. The George Washington dataset comprises 565 text lines from George Washington's letters authored by two 18th-century writers. Bentham's dataset is English scripts in black-and-white images with distorted writing and dark backgrounds. The text in this collection is about 11,500 lines. A sample of Bentham images is shown in Sect. 5.1.2 along with the corresponding preprocessing methods. Each dataset contains 100 characters, including capital and lowercase letters, numerals, punctuation, special symbols, and white space.

4.2 Performance metrics

Character Error Rate (CER) and Word Error Rate (WER) used to evaluate the proposed HTR model. Both equations (CER 5 and WER 6), in all cases, are the total number of insertion, replacement, and deletion operations required to shift from one sequence to another are based on the Levenshtein edit distance [52]. Mainly, CER is defined as

$$\text{CER} = \frac{1}{|N|} \sum_{(p_i, l_i) \in N} \text{LD}(\hat{y}_i, y_i) \quad (5)$$

where $|N|$ is the number of ground truth characters at N partition while the $\text{LD}(\hat{y}, y)$ is Levenshtein distance between prediction \hat{y} and target label y of i th character. In terms of WER , it is defined similarly to CER. Whereas $\text{LD}(\hat{y}, y)$ is computed on word-level that requires to transform one string into another as deletions D word's sum, insertions I and substitutions S by the ground-truth N for that partitions.

$$\text{WER} = \frac{S_{\text{word}} + I_{\text{word}} + D_{\text{word}}}{N_{\text{words}}} \quad (6)$$

4.3 Implementation details

This section describes the implementation of the recurrent baseline model and the proposed transformer-based model.

4.3.1 Baseline recurrent model

This subsection briefly describes the specific implementation details applied to the baseline recurrent model as introduced in the proposed method Sect. 3.1. We stack for both convolution and recurrent layers five blocks each where the convolution layer convolved using 3×3 kernel size. We used zero-padding with stride=1 to maintain the layer's outputs that lead to the exact spatial dimensions as its inputs. A normalization layer (batch norm) is used to help the model converge faster and make it more stable, and a LeakyReLU activation function provides a nonlinearity. Further details, a brief description of the model has depicted in Table 2.

4.3.2 Non-recurrent transformer-based model

The proposed model has a feature size of 2048. The network architecture 3 contains a CNN block for feature extraction, 4 encoder layers, and 4 decoder layers with a hidden dimension of 256. We used a batch size of 16, and ADAM optimizer, and a learning rate of .0006. We adopted the label smoothing technique [37] with Kullback–Leibler divergence cross-entropy

Table 2 Baseline model configuration with the most important hyper-parameters

Layer	Configuration
Input image	1024 × 128 (gray scale)
Convolution layer	16, k: 3 × 3, s: 1, p: same
Batch normalization	
Leaky relu	
Max pooling window	2 × 2, s: 2, p: valid
Convolution Maps	32, k: 3 × 3, s: 1, p: same
Batch norm	
Leaky relu	
Max pooling window	2 × 2, s: 2, p: valid
Dropout	rate: 0.2
Convolution maps	48, k: 3 × 3, s: 1, p: same
Batch Norm	
Leaky Relu	
Max Pooling window	2 × 2, s: 2, p: valid
Dropout	rate: 0.2
Convolution maps	64, k: 3 × 3, s: 1, p: same
Batch Norm	
Leaky Relu	
Self Attention module	
Dropout	rate: 0.2
Convolution Maps	80, k: 3 × 3, s: 1, p: same
Batch Norm	
Leaky Relu	
Self Attention module	
BiLSTM Hidden units1	256
BiLSTM Hidden units2	256
BiLSTM Hidden units3	256
BiLSTM Hidden units4	256
BiLSTM Hidden units5	256
Dropout	rate: 0.5
Decoding	CTC

loss. Every line is padded to the maximum length of 128 characters using a unique character to the right, as illustrated in Fig. 3. Vocabulary size is 100, which include small/capital letter, punctuation marks, numbers, and other special characters.

4.3.3 Kullback–Leibler divergence loss

We calculated that error regularly during the training optimization process to penalize the error model predictions. Choosing a cost function to estimate the model performance allows for updating the weights to minimize the loss on the next epoch. The loss function used in neural network models must fit the issue framing. The softmax activation function generates a probability distribution of over 100 classes

Table 3 The model architecture

Block	Configuration
ResneSt	
2D CNN	(2048 to 256)
Encoder	
Multi-head attention	×4
Normalization layer	×4
Feed forward layer	×4
Normalization layer	×4
Decoder	
Masked multi-head self-attention	×4
Normalization layer	×4
Multi-head attention	×4
Normalization layer	×4
Feed forward layer	×4
Normalization layer	×4
Dense layer	×1
Softmax layer	×1

(multi-class classification problem) in our task. In contrast to the Softmax situation, where the categorical cross-entropy loss function is frequently used, the argmax of the predictions generated compares the predicted distribution with the ground truth distribution. Using Kullback–Leibler Divergence (KLD), [36] is an adaptation of the entropy metric standard in information theory. The predictions generated by the final feedforward layer effectively form a probability distribution and can thus be compared to the actual distribution for the sample x in the corresponding training dataset.

KLD computes the gain and loss of the probability distribution between the predicted distribution of the model $P(t)$ and the distribution of the ground-truth $G(t)$. Backpropagation will continue until the model $P(t)$ produces textual transcription equal to or very similar to the ground truth $G(t)$ distribution probability. The model weights and biases will be adjusted in Eq 7 using the ADAM optimizer to achieve the ideal distribution of the prediction probabilities output.

$$\text{KLD}(P\|G) = - \sum_{i \in X} P(i) * \log \left(\frac{G(i)}{P(i)} \right) \quad (7)$$

where i represents both a sample of the decoded ground truth of X along with its corresponding encoded image feature representation.

4.3.4 Label smoothing

Over-fitting and overconfidence are common issues when training deep learning models. Some regularization techniques have addressed the over-fitting issue like early stop-

Table 4 The impact of attention mechanism using our baseline model on Bentham dataset with no pre-processing technique

Model	CER(%)
CNN-LSTM	11.75
CNN-atten-LSTM	11.10

Bold number shows the influence of attention backbone network against the standard CNN with LSTM decoder, as discussed in Sect. 3.1

ping, weight decay, and dropout. Fortunately, the Label smoothing technique [37] can solve both issues. Equations 8, 9 represent the Label Smoothing that help combine the uniform distribution with updating the one hot coded label vector y that is traditionally used and replacing it with the updated label vector.

$$\hat{y}_i = y_{hot}(1 - \alpha) + \alpha/K \quad (8)$$

where ($K = 100$) is the total number of multi-class categories and y_{hot} represents the embedded ground truth labels.

$$\hat{y}_i = \begin{cases} 1 - \alpha, & i = \text{target} \\ \alpha/K, & i \neq \text{target} \end{cases} \quad (9)$$

In this way, the smoothed distribution of the Label is equivalent to adding noise to the actual distribution to avoid the model being too confident about the correct Label. Therefore, the difference between the output values of the predicted positive and negative samples is not so significant to avoid

overfitting and improve the model's generalization ability. We experimented with a 0.4 as the value of α . Table 3 presents the attention Transformer model architecture.

5 Results and discussions

As mentioned in Sect. 3.3.1, the proposed architecture is based on ResNeSt backbone for feature extraction. All the models were initialized with Imagenet [18] pre-trained weights. We investigate different SOTA neural network models in both: (1) Feature extractions and representation with and without split attention CNN including ResNeSt Image-net pretrained model. (2) Recurrent and non-recurrent encoder–decoder task using both BiLSTM Sect. 3.1 as baseline model, and Transformer encoder–decoder network as proposed model Sect 3.3.2.

5.1 Preliminary results

In the following Sect. (5.1.1 and 5.1.2), we presented the initial results conducted by our experiments on preprocessing methods and the recurrent baseline model. Sect. 5.1.3 presents the result of different backbones for feature extraction in transformer-based model.

5.1.1 Results on baseline model

Table 4 shows the impact of the attention mechanism on the CNN feature extraction method. The CER/WER per-

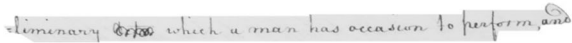
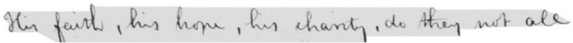
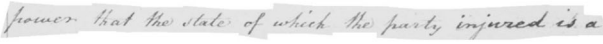

<p style="text-align: center;">a</p>  <p>GT:liminary Acts which a man has occasion to perform, and ResneSt101: liminary Oxtor which a man has occasion to perform, and Resnet101: eiminary aoxt which a man has oasion to perform, and e</p>	<p style="text-align: center;">b</p>  <p>GT:This faith, his hope, his charity, do they not all ResneSt101: -his faith, his hope, his charity, do they not all Resnet101: t-is faith, his hope, his chan -ty, do they not al-e</p>
<p style="text-align: center;">c</p>  <p>GT:power that the state of which the party injured is a ResneSt101: power that the state of which the party injured is a Resnet101: fower that the state of which the partly injured is a</p>	<p style="text-align: center;">d</p>  <p>GT:obstacle might till then have been opposed as such to his escape ResneSt:obstacle might till then have been opposed as such to his excape Resnet:obstacde might till then have been opposed as ruch to his e-cape</p>

Fig. 4 The impact of the Attention mechanism on the prediction of different input images from the Bentham Dataset. The examples are shown from top to bottom: the input image, corresponding ground truth, and

the output prediction with the split attention mechanism is backed by the ResneSt101 CNN. Finally, the output prediction without the attention mechanism is backed by the Resnet101 CNN

Table 5 The impact of preprocessing methods using the proposed baseline attention model on Bentham

Preprocessing	CER(%)
Raw data	11.10
Illumination Compensation (a)	12.03
Deslanted - remove cursive (b)	10.70
both (a) and (b)	11.70

Bold number shows the impact of deslanted algorithm on the preprocessing step as discussed in Sect. 3.2

formance is slightly improved using the baseline-attention model on the Bentham dataset. That leads us to choose the ResNeSt split attention backbone in our proposed model, which increased the performance by 2% in favor of the standard ResNet backbone feature extraction method. Further validation supported the importance of the attention mechanism on CNN are the quantitative and qualitative results reported, as, respectively, demonstrated by Table 7 and Fig. 4 in Sect. 5.1.3. Based on that, we confirm that the CNN-attention-based model captures helpful information, hence becoming robust to the earlier HTR task challenges of handwriting variations. Qualitative results provided visualization evidence where the font red denotes the mispredicted characters over the ground truth and support the quantitative findings showing that the suggested technique is more resilient to inter/intra-class handwriting variances.

5.1.2 The impact of preprocessing methods

To elaborate on our choice of the preprocessing techniques discussed in Sect. 3.2, Table 5 demonstrates the CER on four

training cases. We choose Bentham line text images to investigate our first hypothesis, the better performance of applying massive or light preprocessing techniques. As a result, we found that only using the removal of cursive handwriting (slant removal text method) leads to the lowest CER. Consequently, we employ only the slant removal method for further experiments and discard the illumination compensation techniques on the proposed datasets.

In Fig. 5, we show the sample from the Bentham dataset after reducing the noise, enhancing the lightening, recovering the damaged strokes, and correcting the geometry distortions (correcting the text skew).

As presented in Table 5, we obtain the lowest error rate when only removing the cursive handwriting. As discussed in 3.2, the preprocessing technique employs the de-slanted algorithm to correct the slanted and skewed handwritten text. To elaborate on our choice of the preprocessing techniques discussed in Sect. 3.2, Table 5 demonstrates the CER on four training cases. We choose Bentham line text images to investigate our first hypothesis, the better performance of applying massive or light preprocessing techniques. As a result, we found that only using the removal of cursive handwriting (slant removal text method) leads to the lowest CER. Consequently, we employ only the slant removal method for further experiments and discard the illumination compensation techniques on the proposed datasets. For further experiments, we utilized the removal of cursive handwriting from the images.

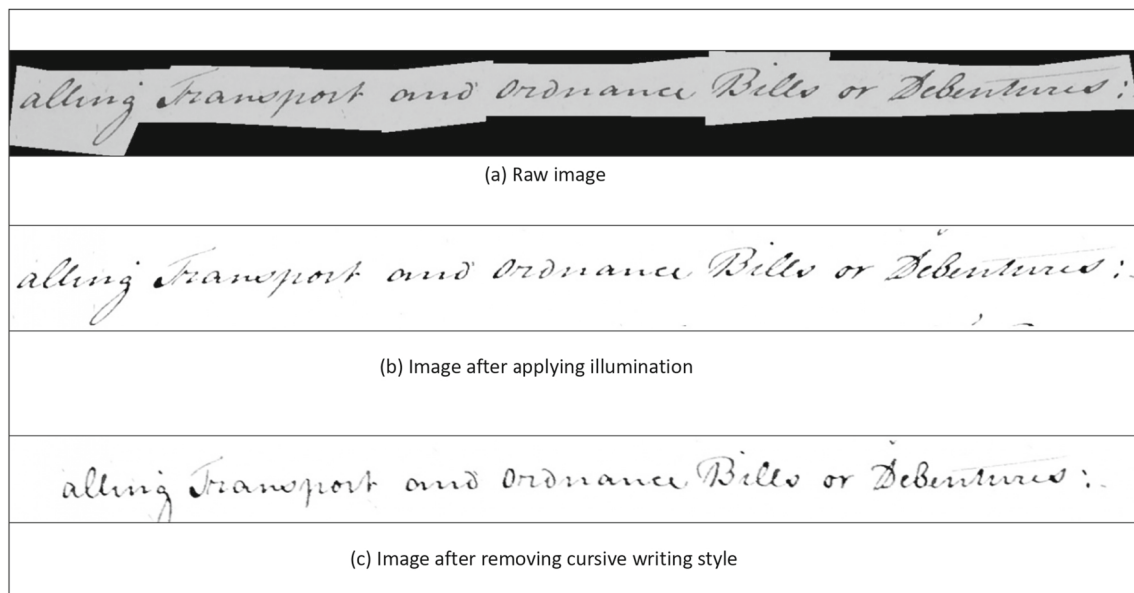


Fig. 5 A sample input image from the Bentham dataset. Top: original image. Middle: an image after applying illumination compensation that cleans any background noise. Bottom: an input image after applying deslanted text removal as pre-processing techniques

Table 6 The impact of deeper layers on Bentham with different backbone feature extraction methods ResNet50 vs ResNet101

Model	CER(%)	WER(%)
ResNet50	7.56	19.01
ResNet101	6.11	14.88

Bold numbers show the best performance with deeper convolutional network (Resnet50 vs Resnet101) as discussed on Sect. 5.1.3

Table 7 The impact of ResNeSt101 attention mechanism on Bentham dataset as feature extraction

Model	CER(%)	WER(%)
ResNet101	6.11	14.88
ResNeSt101	4.90	12.45

Bold numbers show the best accuracy with attention backbone (ResNeSt) over standard convolutional (Resnet) backbone network, as discussed in Sect. 5.1.3

5.1.3 Results on Transformer-based model

After the findings that attention mechanism helps in better feature extraction, we also wanted to experiment with the deeper networks. Hence, we first train Transformer model with ResNet CNNs. Table 6 showcases the results based on ResNets experiment.

Since we obtained better results while using deep layers, we experimented on Bentham dataset to compare the ResNet101 with ResNeSt101 for backbone feature extraction; as anticipated, the latter model provides the more robust feature representation of handwriting images shown in Table 7. Therefore, we considered the ResNeSt as our backbone in the architecture for further experiments, as shown in Fig. 2 and Table 3. The Resnet and ResneSt models perform competitively on the Bentham dataset in the considered standard-CNN and the split attention-CNN approaches. Compared to the Standard CNN-baseline model, the proposed model further decreases both the CER and the WER which can also be observed from the qualitative results reported in Fig. 4 where the reddish color indicates the mis-predicted characters over the ground-truth. The qualitative result shows that the standard backend CNN method has twice CER than the backend CNN with attention ResNeSt. This suggests that, by capturing the more contextual feature representation, the split attention CNN model makes fewer character-level errors and word-level errors with more than 2%. The result of the Bentham historical dataset demonstrates the improvement of model performance when using the attention mechanism.

Further to the importance of the attention mechanism in the feature extraction block, Table 6 presents two experiments on the Bentham dataset. We discovered that the deeper

the network with attention mechanism gets, the better the feature extraction, which ultimately leads to improved accuracy.

5.2 Ablation study

From our findings on previous section, we reported that the network with attention mechanism, the better the feature representation extractions obtained with respect to the best preprocessing choice made by our experiments. We built upon those findings by further improving the SOTA deep learning method for feature extraction and encoder–decoder text transcriber as described in Sect. 3.3.

In ablation research, characteristics from a model are often removed to enhance the model efficiency. The influence on performance is assessed to witness whether or not the model can withstand the removal of specific approaches. This section intensely detailed the impact of transfer learning (TL), data augmentation (DA), and their possible combinations.

5.2.1 An impact of transfer learning

The challenge of training a CNN from scratch is due to different factors, particularly CNN data-hungry and time-consuming. Therefore, TL and DA can solve the HTR task with the shortage of labeled datasets. The Bentham database is excluded from comparing DA with TL since the Bentham dataset plays the source database role in the TL approach, IAM, RIMES, and Washington target datasets. The performance findings of both CER/WER were computed in two scenarios:

- Ensemble approach where we have the average weights of two models trained on 100 and 200 epochs, respectively.
- Best loss where the model records the best validation loss.

Table 8 demonstrates the effect of transfer learning where the source dataset was Bentham. Initially, we trained our model to predict the handwritten text on the Bentham dataset; then, we took the model with the best validation loss for TL on the target datasets, namely IAM, RIMES, and Washington. We also investigated which blocks in ResNeSt should be frozen for best performance. We found out that it is best to train all the blocks. The lowest achieved CER/WER error rates are highlighted in the same Table. It is essential to ensure a reliable comparison with current methods; we used the exact training and testing partitions on a public dataset. The training, validation, and testing procedures used the corresponding set sizes for each dataset are given in partition Table 1.

Table 8 The impact of Transfer Learning where five different scenarios were held on Bentham dataset as the source database concerning the mentioned targeted databases

Freezed blocks	Technique	CER(%)			WER(%)		
		IAM	Rimes	Washington	IAM	Rimes	Washington
All Free	Ensemble	7.48	5.35	6.94	21.61	11.53	19.55
	Best loss	7.49	5.26	7.33	21.60	11.54	20.38
Freeze 1	Ensemble	7.77	5.67	7.36	22.22	11.96	20.59
	Best loss	7.91	5.69	7.17	22.37	11.89	20.19
Freeze 1,2	Ensemble	9.16	6.92	8.58	25.47	12.55	23.44
	Best loss	9.31	6.94	8.88	25.58	12.65	24.08
Freeze 1,2,3	Ensemble	17.18	14.51	21.17	41.11	22.11	47.94
	Best loss	19.61	14.93	20.90	45.10	23.20	48.49
Freeze 1,2,3,4	Ensemble	21.10	17.85	19.14	47.49	25.88	45.52
	Best loss	21.14	17.91	20.59	47.96	25.45	48.81

Bold numbers show the impact of transfer learning from Bentham dataset (large dataset) with 5 different scenarios - where the best findings was with releasing all layers (free) as discussed in Sect. 5.2.1

5.2.2 An impact of both data augmentation and transfer learning

We further analyze the CER/WER performance of both TL and DA techniques when applied to the handwriting system. We followed the same approach as [2]. As they proposed, the alternate techniques (TL and DA) combination has several possible structures. In the first approach, as shown in Table 9, we use DA to learn from the source database (Bentham) and retrain this pre-trained model on the target datasets (IAM, Rimes, Washington).

- The model is trained from scratch using a source dataset augmented with data.
- The model is retrained using target datasets after the data augmentation technique has been applied to the source dataset

this is referred to as the DA-TL-DA paradigm.

On the other hand, in the second proposal, we used the DA technique but did not apply it to the target datasets:

- The model is trained from scratch using a source Bentham dataset using the DA technique.
- the model is retrained without using DA on the target datasets.

To be thorough, in Table 9, we report the findings on the proposed architecture with all block layers unfrozen. The baseline model is the CNN-Transformer model with no techniques applied to it. The DA-TL and DA-TL-DA techniques use the Bentham database to train the model from scratch. The model is then calibrated using data from the IAM, Rimes, and Washington databases, with/without augmentation on the target datasets. As imposed to [2], we can conclude that applying DA over the target datasets after TL is applied, i.e.,

the DA-TL-DA paradigm, does improve the CER and WER performance. In the case of the Rimes dataset, the DA-TL-DA approach did not perform well; one possible reason for this poor performance could be that the French language on the target dataset is slightly different from the source Bentham English dataset. We acknowledged that applying DA on target datasets following TL is advantageous. The DA-TL-DA paradigm is beneficial when the target source has only a few textually labeled lines, as with the Washington database. After considering the arguments above and Tables 8 and 9, it can be concluded that the DA-TL-DA technique is reliable. The starting point is relatively good when fine-tuning a ResNeSt trained on a similar task, such as the Bentham dataset as an extensive database of HTR samples. Without additional training on the target datasets, we demonstrate that the model can provide good generalization for the DA as in the Rimes dataset and DA+TL-DA as in the rest target datasets. The proposed model is then trained against the target databases containing only a few input samples, as in the Washington database, representing only a tiny portion of the training set.

5.3 Comparison with the state-of-the-art on benchmark datasets

In Table 10, we present a comprehensive performance comparison with the state-of-the-art. Different approaches have been compared to our work depending on whether they required a pre-segmentation task, known as segmentation-based, or not, known as segmentation-free, and whether they required a language model (lexicon-based) or not (lexicon-free). Some techniques are based on recurrent neural networks, typically LSTMs with a Connectionist Temporal Classification (CTC) loss function or Transformer encoder-decoder sequence-to-sequence architectures.

Table 9 The impact of Transfer Learning and data augmentation where five different scenarios were held on Bentham dataset as the source database concerning the mentioned targeted databases

Method	Technique	CER(%)			WER(%)		
		IAM	Rimes	Washington	IAM	Rimes	Washington
Baseline	Ensemble	8.71	4.51	72.59	25.31	12.82	90.68
	Best loss	8.97	4.59	75.79	25.66	12.92	91.39
+TL	Ensemble	7.48	5.35	6.94	21.61	11.53	19.55
	Best loss	7.49	5.26	7.33	21.60	11.45	20.38
+ DA	Ensemble	7.85	4.24	70.81	21.46	11.04	88.85
	Best loss	7.77	4.25	73.68	21.46	10.03	89.78
+ DA+TL	Ensemble	6.98	4.86	5.76	19.50	13.85	16.36
	Best loss	7.10	4.66	6.01	19.90	13.41	16.38
+ DA+TL+DA	Ensemble	6.66	5.53	4.84	18.97	12.48	13.75
	Best loss	6.70	5.42	4.74	18.90	11.98	13.23

Bold numbers show the best performance on our model while carrying out both transfer learning and data augmentation as discussed in Sect. 5.2.2

Table 10 Results of text recognition on four benchmark databases: IAM, Rimes, Washington, Bentham datasets

Model	CER(%)				WER(%)			
	IAM	Rimes	Wash.	Ben.	IAM	Rimes	Wash.	Ben.
Chen et al. [14]	11.15	8.29	–	–	34.55	30.54	–	–
Pham et al. [39]	13.92	8.62	–	–	31.48	27.01	–	–
Wigington, et al. [49]	6.07	3.09	17.50	–	19.07	11.29	65.20	–
Aradillas et al. [2]	5.90	2.70	18.70	–	20.30	10.70	69.20	–
Bluche et al. [6]	3.20	1.90	–	–	10.50	7.90	–	–
Kang et al. [30]	4.67	–	–	–	15.45	–	–	–
Flor et al. [17]	8.52	6.78	–	8.49	30.58	29.75	–	24.04
Puigcerver et al. [43]	9.32	8.56	–	8.16	32.13	31.29	–	24.43
Cascianelli et al. [9]	6.8	4.0	–	–	24.7	13.7	–	–
Ours	6.66	4.24	4.74	4.90	18.97	10.03	13.23	12.45

Bold numbers present the best model performance on different datasets with comparison to other related models as discussed in Sect. 5.3

Our results are compared against the nine related studies to demonstrate the improvement and evaluate the accuracy of the proposed model. These approaches used deep learning methods based on CNN and LSTM-CTC or Transformer encoder–decoder, where some variant of the LSTM is used. Some studies like using DA, synthetic datasets in the target datasets, and Lexicon Model (LM). We track the percentage of characters and words incorrectly identified as CER and WER, respectively. The proposed model is tested on four benchmark datasets using their standard partitions as described in our experiments in Sect. 4.1, to ensure a fair comparison with the state-of-the-art methods in Table 10. Additional experimental protocols used in the existing studies to further compare our work are: (1) recurrence or nonrecurrence approaches, (2) employed DA-TL or not, (3) lexicon-based or lexicon-free model, and (4) using synthetic dataset or not. Though Chen et al. [14] attempt to provide a multi-HTR systems-based recurrent network, their model poorly performed on IAM and Rimes compared to our finding. Although Pham et al. [39] proposed and empowered their

model with Dropout and a constrained language and lexicon-based model, their model performance remains farther than ours. Likewise, in our proposed model, Wigington et al. [49] applied DA to their studies, and we got close CER/WER in IAM and Rimes datasets. However, in the Washington dataset, we outperform their approach with more than 12%. Similarly, Aradillas et al. [2] outperform our (DA+TL+DA) paradigm approach, with a slight value of less than 1% CER on IAM and around 2% CER on the RIMES dataset. However, we obtain better WER performance on those datasets as well as on the Washington dataset. Bluche et al. [6] and Kang et al. [30] outperform the proposed model on IAM and Rimes with some constraints on their models like generating synthetic datasets and applying language model. However, the improvement is significantly noticed using the transformer as the decoder part; unlike our work, Bluche and Kang boost their performance with the help of generating a synthetic dataset and using the lexicon-based model. Finally, our proposed model marginally outperforms the performance of the work of Flor et al. [17], Puigcerver et al. [43] and Cas-

Table 11 The impact of CLIP model pre-training with Augmentation on IAM dataset

Method	Technique	IAM	
		CER(%)	WER(%)
No Pre-training	Ensemble	7.85	21.46
	Best loss	7.77	21.30
Pre-training	Ensemble	7.65	21.36
	Best loss	7.59	21.23

cianelli et al. [9]. However, Puigcerver used DA and synthetic datasets to empower their performance model. In contrast, the work of Cascianelli employed deformable convolutions that resulted in competitive CER/WER in IAM and Rimes datasets.

During the evaluation process, we also try to use Clip Pre-training on the IAM dataset. Since we aimed to discover the effect of CLIP [44] pre-training the backbone CNN model on the performance of HTR's CER since the CLIP is exceptionally well-known for its capabilities these days. We decided to use the Clip-pretraining on the ResNeSt model. The approach is similar to what is described in the CLIP model. The ground truth labels are fed to the text encoder sub-encoder model and output a list of their corresponding textual embedding, called mathematical space. On the other hand, the equivalent input images fed to the sub-encoder model output its relative feature representations that serve as input for the Transformer encoder.

After pre-training the CNN model on the IAM dataset, we used the same weights to train the CNN-Transformer model on the HTR task. Table 11 shows the impact of clip training on the CER for the IAM dataset. DA is done on both of the experiments. We can conclude that pre-training the CNN model helps improve the model's performance.

6 Conclusion

This paper presents a transformer-based and lexicon-free approach for HTR. To the best of our knowledge, this is the first approach to the HTR task that utilizes a joint attention mechanism on both feature extraction (CNN) and encoder-decoder text transcriber (transformer) networks. We examined different preprocessing methods before feeding data to the purposed model; we found that removing only the slanted text technique improves performance. We conducted a rigorous analysis and evaluation of several experiments with (TL) and (DA) paradigms, demonstrating DA+TL+DA paradigm opted to be the proposed and optimal design for the HTR task. Our findings on the reported results show that our method can achieve the best possible results (SOTA) with neither synthetic data nor a lexicon model. Also, our proposed

model can deal with small-shot training datasets such as the Washington dataset, extending its relevance to real-world use cases. Transformers are excellent at integrating visual and language-specific knowledge since they are character-based rather than vocabulary-based.

More specifically, the HTR problem is a generic text recognition task. Therefore, we provided an end-to-end neural network model trained on variable-sized line images with their corresponding line-level transcriptions. To summarize our contributions on the following lines:

- We studied the impact of the attention mechanism on CNN-backbone feature extraction
- We examined the influences of the prior utilizing different preprocessing techniques.
- We devised a unified deep learning framework from SOTA dual attention mechanism networks
- We comprehensively investigated the effect of TL and DA.
- Our proposed model with lexicon-free and no synthetic data outperforms the performance of the cited SOTA models with the same constraints.

We completed thorough evaluations on four public benchmark datasets for handwriting text recognition. We achieved SOTA performance utilizing the identical architecture and the minimum hyper-parameter changes. That makes our model more robust, universal, and straightforward for any new text recognition challenge. Extensive experiments are performed on four public datasets, and both the source code and all of the pre-trained models will be available on our GitHub.

In the future, we aim to improve the architectural design by incorporating (TL) and (DA) with the CLIP model to decrease the CER/WER further. In addition, synthesizing good enough handwriting text lines will help the model train on a substantial amount of data that improve the model performance and addresses overfitting. The closed vocabulary (LM) should incorporate the DA-TL-DA and CLIP techniques to improve CER/WER.

Acknowledgements The authors would like to thank NSERC Canada for their financial support under grant # 2019-05230.

References

1. Aberdam, A., Litman, R., Tsiper, S., Anshel, O., Slossberg, R., Mazor, S., Manmatha, R., Perona, P.: Sequence-to-sequence contrastive learning for text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15302–15312 (2021)
2. Aradillas, J.C., Murillo-Fuentes, J.J., Olmos, P.M.: Boosting offline handwritten text recognition in historical documents with few labeled lines. *IEEE Access* **9**, 76674–76688 (2021)

3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2014). arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473)
4. Bhunia, A.K., Khan, S., Cholakal, H., Anwer, R.M., Khan, F.S., Shah, M.: Handwriting transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1086–1094 (2021)
5. Bianne-Bernard, A.L., Menasri, F., Mohamad, R.A.H., Mokbel, C., Kermorvant, C., Likforman-Sulem, L.: Dynamic and contextual information in hmm modeling for handwritten word recognition. *IEEE Transact. Pattern Anal. Mach. Intell.* **33**(10), 2066–2080 (2011)
6. Bluche, T., Messina, R.: Gated convolutional recurrent neural networks for multilingual handwriting recognition. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 646–651. IEEE (2017)
7. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. *Adv. Neural Inform. Process. Syst.* **33**, 1877–1901 (2020)
8. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229. Springer (2020)
9. Cascianelli, S., Cornia, M., Baraldi, L., Cucchiara, R.: Boosting modern and historical handwritten text recognition with deformable convolutions. *International Journal on Document Analysis and Recognition (IJDAR)* 1–11 (2022)
10. Causer, T., Wallace, V.: Building a volunteer community: results and findings from Transcribe Bentham. *Digit. Humanit. Q.* **6**(2) (2012)
11. Chammas, E., Mokbel, C., Likforman-Sulem, L.: Handwriting recognition of historical documents with few labeled data. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), pp. 43–48. IEEE (2018)
12. Chen, K.N., Chen, C.H., Chang, C.C.: Efficient illumination compensation techniques for text images. *Digital Signal Process.* **22**(5), 726–733 (2012)
13. Chen, X., Mishra, N., Rohaninejad, M., Abbeel, P.: Pixelsnail: An improved autoregressive generative model. In: International Conference on Machine Learning, pp. 864–872. PMLR (2018)
14. Chen, Z., Wu, Y., Yin, F., Liu, C.L.: Simultaneous script identification and handwriting recognition via multi-task learning of recurrent neural networks. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 525–530. IEEE (2017)
15. Cheng, J., Dong, L., Lapata, M.: Long short-term memory-networks for machine reading (2016). arXiv preprint [arXiv:1601.06733](https://arxiv.org/abs/1601.06733)
16. Cui, Z., Ke, R., Pu, Z., Wang, Y.: Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction (2018). arXiv preprint [arXiv:1801.02143](https://arxiv.org/abs/1801.02143)
17. de Sousa Neto, A.F., Bezerra, B.L.D., Toselli, A.H., Lima, E.B.: Htr-flor++ a handwritten text recognition system based on a pipeline of optical and language models. In: Proceedings of the ACM Symposium on Document Engineering 2020, pp. 1–4 (2020)
18. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
19. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018). arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
20. Dong, L., Xu, S., Xu, B.: Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5884–5888. IEEE (2018)
21. Espana-Boquera, S., Castro-Bleda, M.J., Gorbé-Moya, J., Zamora-Martínez, F.: Improving offline handwritten text recognition with hybrid hmm/ann models. *IEEE Transact. Pattern Anal. Mach. Intell.* **33**(4), 767–779 (2010)
22. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376 (2006)
23. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. *IEEE Transact. Pattern Anal. Mach. Intell.* **31**(5), 855–868 (2008)
24. Graves, A., Schmidhuber, J.: Offline handwriting recognition with multidimensional recurrent neural networks. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) *Advances in Neural Information Processing Systems*, vol. 21, pp. 545–552. Curran Associates, Inc. (2008). <https://proceedings.neurips.cc/paper/2008/file/66368270ffd51418ec58bd793f2d9b1b-Paper.pdf>
25. Gregor, K., Danihelka, I., Graves, A., Rezende, D., Wierstra, D.: Draw: A recurrent neural network for image generation. In: International Conference on Machine Learning, pp. 1462–1471. PMLR (2015)
26. Grosicki, E., Carré, M., Brodin, J.M., Geoffrois, E.: Results of the rimes evaluation campaign for handwritten mail processing. In: 2009 10th International Conference on Document Analysis and Recognition, pp. 941–945. IEEE (2009)
27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
28. Kane, S., Lehman, A., Partridge, E.: Indexing george washington's handwritten manuscripts. Center for Intelligent Information Retrieval, Computer Science Department, University of Massachusetts, Amherst, MA 1003 (2001)
29. Kang, D., Lv, Y., Chen, Y.Y.: Short-term traffic flow prediction with lstm recurrent neural network. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6. IEEE (2017)
30. Kang, L., Riba, P., Rusiñol, M., Fornés, A., Villegas, M.: Pay attention to what you read: Non-recurrent handwritten text-line recognition (2020). arXiv preprint [arXiv:2005.13044](https://arxiv.org/abs/2005.13044)
31. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)
32. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
33. Marti, U.V., Bunke, H.: The iam-database: an english sentence database for offline handwriting recognition. *Inter. J. Doc. Anal. Recognit.* **5**(1), 39–46 (2002)
34. Mermelstein, P., Eyden, M.: A system for automatic recognition of handwritten words. In: Proceedings of the October 27–29, 1964, fall joint computer conference, part I, pp. 333–342 (1964)
35. Michael, J., Labahn, R., Grüning, T., Zöllner, J.: Evaluating sequence-to-sequence models for handwritten text recognition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1286–1293. IEEE (2019)
36. Moreno, P., Ho, P., Vasconcelos, N.: A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. *Adv. Neural Inf. Process. Syst.* **16** (2003)
37. Müller, R., Komblith, S., Hinton, G.E.: When does label smoothing help? In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F. (eds.) *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates,

- Inc. (2019). <https://proceedings.neurips.cc/paper/2019/file/f1748d6b0fd9d439f71450117eba2725-Paper.pdf>
38. Parikh, A.P., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference (2016). arXiv preprint [arXiv:1606.01933](https://arxiv.org/abs/1606.01933)
39. Pham, V., Bluche, T., Kermorvant, C., Louradour, J.: Dropout improves recurrent neural networks for handwriting recognition. In: 2014 14th International Conference on Frontiers in Handwriting Recognition, pp. 285–290. IEEE (2014)
40. Plizzari, C., Cannici, M., Matteucci, M.: Skeleton-based action recognition via spatial and temporal transformer networks. *Comput. Vision Image Understanding* **208**, 103219 (2021)
41. Plötz, T., Fink, G.A.: Markov models for offline handwriting recognition: a survey. *Inter. J. Doc. Anal. Recognit. (IJDAR)* **12**(4), 269–298 (2009)
42. Poznanski, A., Wolf, L.: Cnn-n-gram for handwriting word recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2305–2314 (2016)
43. Puigcerver, J.: Are multidimensional recurrent layers really necessary for handwritten text recognition? In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 67–72. IEEE (2017)
44. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
45. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
46. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transact. Pattern Anal. Mach. Intell.* **39**(11), 2298–2304 (2016)
47. Sueiras, J., Ruiz, V., Sanchez, A., Velez, J.F.: Offline continuous handwriting recognition using sequence to sequence neural networks. *Neurocomputing* **289**, 119–128 (2018)
48. Vinciarelli, A., Luetin, J.: A new normalization technique for cursive handwritten words. *Pattern Recognit. Lett.* **22**(9), 1043–1050 (2001)
49. Wigington, C., Stewart, S., Davis, B., Barrett, B., Price, B., Cohen, S.: Data augmentation for recognition of handwritten words and lines using a cnn-lstm network. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 639–645. IEEE (2017)
50. Yadav, S.S., Jadhav, S.M.: Deep convolutional neural network based medical image classification for disease diagnosis. *J. Big Data* **6**(1), 1–18 (2019)
51. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 21–29 (2016)
52. Yujian, L., Bo, L.: A normalized levenshtein distance metric. *IEEE Transact. Pattern Anal. Mach. Intell.* **29**(6), 1091–1095 (2007)
53. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: International Conference on Machine Learning, pp. 7354–7363. PMLR (2019)
54. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., et al.: Resnest: Split-attention networks (2020). arXiv preprint [arXiv:2004.08955](https://arxiv.org/abs/2004.08955)
55. Zhang, Y., Nie, S., Liu, W., Xu, X., Zhang, D., Shen, H.T.: Sequence-to-sequence domain adaptation network for robust text image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2740–2749 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.