An End-to-End Approach for Handwriting Recognition: From Handwritten Text Lines to Complete Pages

Dayvid Castro
Universidade Federal de Pernambuco
Recife, Pernambuco, Brazil
dwco@cin.ufpe.br

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

Cleber Zanchettin
Universidade Federal de Pernambuco
Recife, Pernambuco, Brazil
cz@cin.ufpe.br

Abstract

Handwritten Document Recognition (HDR) has emerged as a challenging task integrating text and layout information recognition to tackle manuscripts end-to-end. Despite advancements, the computational efficiency of processing entire documents remains a critical challenge, limiting the practical applicability of these models. This paper presents the Document Attention Network for Computationally Efficient Recognition (DANCER). The model differs from existing approaches with its unique encoder-decoder structure, where the encoder reduces spatial redundancy and enhances spatial attention, and the decoder, comprising transformer layers, efficiently decodes the text using optimized attention operations. This design results in a fast, memory-efficient model capable of effectively transcribing and understanding complex manuscript layouts. We evaluated DANCER's efficacy on the ICFHR 2016 READ competition dataset, focusing on recognizing single and double-page historical documents. We demonstrate how DANCER can triple the training batch size compared to prior models within the same memory limits and reduce memory usage by up to 65% without compromising recognition quality. The proposed approach sets new standards in efficiency and accuracy for HDR solutions, paving the way for practical and scalable applications in diverse contexts.

1. Introduction

Despite the advent of digital technologies, handwriting remains deeply rooted in various aspects of our lives, from personal notes and letters to official documents and historical archives that preserve our society's written past. The ability to automatically transcribe and analyze handwritten content allows information retrieval and preservation, in addition to enabling the development of intelligent systems that can interact with and understand manuscript content.

Handwritten Text Recognition (HTR) is a challenging research area with numerous applications that range from digitizing historical documents to automatic handwritten data ingestion. Over the years, deep learning models have achieved remarkable performance in HTR by leveraging recurrent and convolutional neural networks [34, 43], and, more recently, attention-based models [10, 11, 28].

Recent advances in the handwritten text recognition field based on transformer models [12, 13] enabled the recognition of whole documents in a segmentation-free end-to-end fashion without constraints present in previous works [10, 11], such as relying on segmented text regions. Removing the segmentation step avoids the possibility of errors stemming from this stage, simplifies HTR pipelines, and increases the available context for learning and interpretation. The so-called Handwritten Document Recognition (HDR) is an exciting modeling framework that recognizes both the handwritten content and layout entities to tackle manuscripts end-to-end.

The current approaches for HDR present high computational costs, mainly arising from the use of transformer layers with quadratic time and memory complexity in sequence length [17]. Yet, there are multiple benefits to improving the computational cost of handwritten text recognition models. Minimizing the memory requirements of HTR models not only impacts their practical use in real-world scenarios but also makes them more accessible to the general public, broadening the potential applications of these technologies. Besides, it allows training and deploying these models

in limited hardware with low memory resources and favors larger batch sizes that decrease the total training time.

Optimizing the time required to train new models can be crucial for research and industrial parties as it helps to innovate at a faster pace. There are also environmental concerns, as optimizing training time helps minimize the carbon footprint and energy consumption. Furthermore, a faster solution means quicker iterations, enabling researchers and practitioners to explore various architectures and hyperparameters, which can lead to reduced development time.

In addition to the speed and memory concerns, maintaining high recognition rates is essential to delivering meaningful transcriptions. Accordingly, our primary goal is to propose a faster document recognition model that requires less memory and exhibits recognition rates on par with state-of-the-art (SOTA) approaches. With this goal in mind, we present a Document Attention Network for Computationally Efficient Recognition (DANCER). Compared to SOTA models, our HDR model significantly improves computational efficiency without sacrificing recognition accuracy.

We conducted extensive experiments on a historical handwriting benchmark and compared DANCER with existing state-of-the-art models. The results suggest that DANCER presents superior performance in speed and memory requirements while maintaining robust recognition rates that are on par with SOTA. We evaluated DANCER in single-page and double-page versions of the READ 2016 historical dataset. Among the improvements obtained, we can highlight that DANCER can process three times more samples than previous solutions with a memory reduction of up to 65% compared to other methods under similar conditions. We can summarize our contributions as follows:

- We propose a new model for handwritten document recognition that significantly reduces computational costs, offering a novel approach regarding efficiency and scalability. The key innovations that our model introduces for the document recognition task include:
  - Enhancement of the encoder recognition module with a gated depth-wise separable convolution for advanced attention-aware feature extraction, enabling selective focus on key elements within handwritten texts.
  - The use of an optimized convolution operation, called octave convolution [9], that minimizes the spatial redundancy, thus enhancing processing efficiency.
  - Enhancement of the decoder module through highly efficient attention operations, i.e., memory-efficient attention and FlashAttention [14, 35], to address computational limitations of traditional attention models.

  Collectively, these changes make our approach more efficient and scalable to larger and more complex datasets.
- We carried out thorough evaluations of DANCER and predecessor models in both single and double-page document recognition tasks, demonstrating the adaptability and effectiveness of our model across multiple real-world scenarios. The proposed model excels in processing efficiency and memory utilization while preserving high recognition accuracy.
- We perform scalability analyses that help assess the benefits brought by our proposal. We discover that DANCER is better at managing the computational resources as the compute load grows, whether in terms of batch size or the document's number of text lines.

2. Related Work

Most of the research in HTR has been conducted based on solutions requiring segmented characters, words, lines, paragraphs, or an isolated text column. That implies that layout analysis and segmentation are necessary before text recognition. However, over the past few decades, there has been a trend toward minimizing the restrictions required to recognize handwritten texts, aiming to build what is referred to as unconstrained handwritten text recognition. As such, removing the segmentation requirements has been the target of many previous works since the step might cause subsequent errors in a recognition pipeline.

In this context, we can highlight the method proposed by [19] to directly recognize text lines without explicit character segmentation using the Connectionist Temporal Classification (CTC) objective function. The use of the CTC loss to train recurrent or convolutional neural networks became a standard practice to recognize lines of handwritten text until the present [3, 7, 8, 16, 21, 22, 30, 34, 38, 41, 43]. Following a different line, other researchers have investigated attention models to predict character sequences from text line images [1, 23, 28, 33, 44].

There were also developments toward recognizing handwritten content within paragraph images, meaning an explicit line segmentation step would not be needed [2, 4, 37]. Initial works such as the ones by [2, 4] used Multi-dimensional Long-Short Term Memory (MDLSTM) networks to build attention-based models. However, the results were far from what was achieved with line-level solutions. [4] used a character-based MDLSTM attention model with speed limitations mainly due to the decoding processing recognizing a single character at a time.

More recent studies such as [10, 11] matched or outperformed systems that receive pre-segmented lines working directly on paragraph images. On the one hand, [11] proposes the Vertical Attention Network (VAN) approach, an encoder-decoder architecture with an attention mechanism for selecting the features that represent the current text line being read combined with a Long-Short Term Memory network. On the other hand, the model proposed in [10] is a recurrence-free and attention-free alternative to the previous version, having a single convolutional layer as the decoder.

Although the proposals in [10, 11] can perform the

recognition without explicitly segmenting the text lines, the investigations are restricted to paragraph images. A natural evolution is, therefore, the recognition of complete text pages. Under this challenge, the first approaches were two-step solutions that would first detect the text lines and then proceed to their recognition. [31] introduced a model that learns to predict Start of Line (SoL) positions to derive a bounding box to extract the line and feed a line-level HTR model. In the Start-Follow-Read (SFR) model [45], besides learning to predict the SoL, a line follower network assists in extracting warped lines to feed a line-level recognition module. Suit et al. [40] jointly train text line detection and recognition neural networks under a multi-task learning framework and take advantage of weight sharing between both models to increase performance. Other works also follow a similar strategy of jointly training text detection and transcription networks [5, 6].

Recent works [39, 46] presented solutions that avoid the two-step segmentation and line-level recognition framework. Instead, these models directly learn to recognize manuscript content without segmenting lines or paragraphs. [46] achieves full-page recognition by unfolding the input paragraph image into a single very large text line, with the downside that it does not preserve the line breaks. Since it recognizes a continuous list of characters over a flattened 1D vector, it only handles pages with one column due to the lack of understanding of the document layout.

In [39], an attention-based encoder-decoder model capable of handling complex document pages such as two columns is presented. The authors used the ResNet-34 [20] backbone for the encoder and a transformer network as the decoder [42]. The model has approximately 28 million parameters, considerably higher than other works, including ours. Although the model has fewer constraints compared to early works, the error rates are still significantly below what is achieved with cropped text line approaches. As a result of the developments aiming to build a robust and unconstrained handwritten text recognition solution, a recent trend refers to recognizing both transcription and layout entities contained in a document as a unified task.

2.1. Handwritten Document Recognition (HDR)

Handwritten Document Recognition is an evolving field that integrates layout understanding with textual recognition, enabling end-to-end processing of manuscripts. The shift from recognizing isolated text regions to entire documents encompasses complexities that span spatial, contextual, and structural dimensions. HDR must contend with diverse document layouts that may include multiple columns and annotations. This complexity requires sophisticated models to distinguish between different text regions and understand their spatial relationships.

The Document Attention Network (DAN) [12] is the first model to tackle this task with an encoder-decoder model based on a vanilla convolutional encoder followed by a traditional transformer decoder [42]. The model is trained using cross-entropy loss on a softmax output layer with units representing the respective language alphabet and the layout entities. [29] evaluates the DAN model to recognize Russian manuscript documents. One of the main drawbacks of DAN is its high computational cost. Considering the time aspect, the decoding process is conducted one character at a time, so the inference time for a whole page is substantial.

Faster DAN [13] is an evolution model that improves the inference speed through a multi-line positional encoding strategy that allows us to decode characters from different lines in a parallel manner. However, there are computing limitations even in this optimized model. Using traditional transformers comes with inherent computational cost challenges, e.g., high memory consumption, training time, and scaling issues [14, 24, 35], due to the quadratic complexity of the standard attention [42]. In light of these challenges, our work seeks further improvements in the computing side by proposing a model design with reduced computational complexity to obtain a faster and less memory resource-intensive solution without compromising recognition accuracy.

3. Proposed DANCER Model

Our document recognition pipeline starts with a preprocessing stage that consists of a standardization step to normalize the input. During the model training, preprocessing includes on-the-fly data augmentation comprising a set of techniques commonly adopted in predecessor studies [12, 13] (random scale, random perspective, elastic distortion, dilation and erosion, color jittering, gaussian noise, gaussian blur, and random sharpening). The optical model DANCER receives the preprocessed document image and end-to-end recognizes the manuscript content and layout information.

3.1. Problem Formulation

We formulate the HDR task as finding the likeliest token sequence $t^* = (t_1 ... t_n)_{t \in \mathcal{T}}$ given the document image D, where $\mathcal{T} = \mathcal{A} \cup \mathcal{L} \cup \{< \text{eot} >\}$ encompasses the dataset's alphabet $\mathcal{A}$, the layout tokens $\mathcal{L}$ and an extra token $< \text{eot} >$ that indicates when the model has finished the transcription. Layout tokens are in the form of XML tags with an open and a closed variant.

3.2. Architecture

The architecture of the computationally efficient document attention network we introduce in this paper is illustrated in Figure 1. The network follows an encoder-decoder design composed of an optimized fully convolutional encoder and transformer decoder.

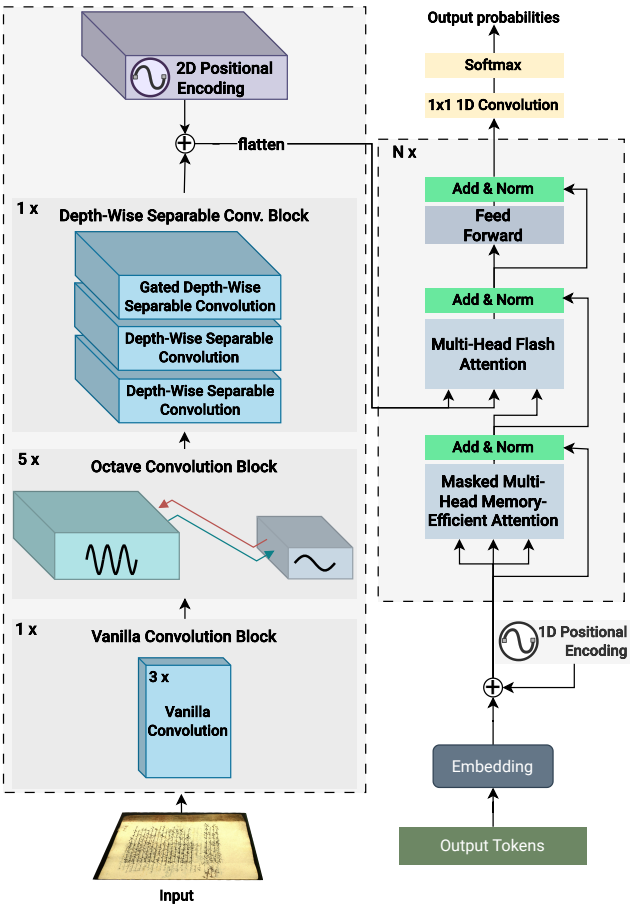


Figure 1. Architecture of the Document Attention Network for Computationally Efficient Recognition. The model follows an encoder-decoder design. The encoder comprises vanilla, octave, depthwise, and gated conv layers. The decoder constitutes transformer layers based on memory-efficient and FlashAttention.

The encoder was designed to be faster while still being capable of learning a representative space that captures the important information from the documents. It comprises ReLU-activated convolution layers of different types combined with max pooling, normalization, and dropout layers. Due to the benefits found in [11–13], we use a Diffused Mix Dropout [11] to increase regularization. It consists of varying the layer's localization given a predefined set of available positions and the type of dropout layer – standard or spatial dropout. The localization and dropout types are randomly selected. We use 0.5 and 0.25 dropout probabilities for standard and 2D dropout layers.

The encoder starts processing the document image with a vanilla convolutional block. The block comprises three traditional convolution layers plus an instance normalization layer to increase the model's stability. The Mix Dropout is placed before the first or second convolution layer in this first block.

The vanilla's convolutional block feeds a sequence of five blocks of octave convolutions (OctConv), comprising most of the encoder layers. OctConv is a special convolutional layer that splits the signal into high and low-frequency paths to decrease the redundancy of slow-changing features stored under the same spatial resolution as the high-frequency features.

We chose this convolution because it can deliver a robust feature learning capability while improving speed and memory consumption. This multi-scaling strategy might be beneficial in capturing features on different abstraction levels. That is, it could enable the optical model to capture the complex details of character shapes, such as thin lines and character stroke combinations (through the high-frequency path), as well as the overall patterns and standard text structures (through the low-frequency path), which are imperative for understanding the text's comprehensive layout and contextual background.

The octave blocks consist of two octave convolution layers followed by an instance normalization layer. We also include the Diffused Mix Dropout and the possible locations where it can be applied are before the second convolution layer, before the normalization layer, or after the normalization layer.

The last block comprises three highly efficient depthwise separable conv layers. The adoption of this convolution type in the upper layers, which holds the highest number of filters, is strategically motivated by its efficiency benefits. Depth-wise separable convolution, by decoupling the spatial and depth (channel) dimensions of the convolution process, significantly reduces the computational complexity and the number of parameters compared to traditional convolution.

We introduce a gated convolution [15] in the depth-wise portion of the last layer to include spatial attentional awareness in our encoder. The gated conv operates by applying an element-wise multiplication of the convolution output with a gating signal. The signal is generated by another sigmoid-activated convolution, enabling selective focus on the more relevant features. The gated depth-wise separable conv can potentially help our model focus on the relevant handwriting content and avoid the noise usually present in historical documents.

The sequence of encoder blocks totalizes 16 layers plus the extra gated convolution inside the last gated depth-wise separable convolution. For detailed hyper-parameter information, refer to the supplementary material. Next, a series of 8 transformer decoder layers receive a flattened version of the convolutional encoder's output with a 2D positional encoding strategy to maintain information about the document image's multi-dimensional nature. The number of attention heads is set to 4, and the dropout to 0.1. The outcomes of previous studies directly influenced these hyper-

parameter settings [12, 13]. The output token embeddings are also encoded with positional data and provided to the decoder. We employed the multi-query positional encoding strategy devised in [13] to improve inference time by decoding multiple characters simultaneously. This is enabled by a two-step decoding process that first predicts the start of each line and then simultaneously queries for multiple line token outputs.

The general structure of the decoder layer follows the original transformer architecture comprising a masked self-attention layer, a cross-attention layer, and a feed-forward neural network block. Residual connections follow all three components. We integrate two optimized attention operations—memory-efficient attention [35] for self-attention layers and FlashAttention [14] for the cross-attention layers—to harness the combined benefits of these two advanced mechanisms within DANCER's architecture. The design's philosophy behind memory-efficient attention focuses on reducing the total memory footprint, while FlashAttention focuses on decreasing the number of memory reads and writes to improve runtime. This approach was motivated by the need to enhance computational efficiency and manage the model's scalability when handling the complex task of handwritten document recognition. Our goal is to ensure that our model remains robust and responsive even as the size of the documents increases.

4. Experiments

4.1. Data and Metrics

READ Statistics	Single Page	Double Page
Training Samples	350	169
Validation Samples	50	24
Test Samples	50	24
Total	450	217
#Chars	89	89
#Layout Tokens	10	10
Width (pixels)	1198.75	2396.48
Height (pixels)	1761.55	1761.58

Table 1. Details of the experimental data used. The values of image width and height are averaged values across the dataset.

The experiments were conducted over the ICFHR 2016 READ dataset [36]. The data, derived from the state archive of Bozen and part of the European Union's Horizon 2020 READ project, consists of documents from the Ratsprotokolle collection – based on minutes of council meetings from 1470 to 1805. It is composed of Early Modern German handwriting and was introduced in the ICFHR 2016 competition for handwritten text recognition of historical documents. READ is organized with a classification system for text regions into five classes: page (P), page number (N), body (B), annotation (A), and section (S). An automated process establishes the reading order, prioritizing page numbers followed by sections (with annotations read before the body). READ is divided into training, validation, and test sets. We use the official single-page partition and the double-page setup in [12, 13]. Table 1 presents details of the data collection.

We compute metrics to evaluate the recognition quality of both the textual transcriptions and the layout understanding. We measure the text recognition performance using the well-stabilized metrics Character Error Rate (CER) and Word Error Rate (WER) based on the edit distance [27]. On the layout side, we use the Layout Ordering Error Rate (LOER) [12], which models the document layout as an oriented graph. This approach allows for a detailed assessment of layout recognition by considering both the structural hierarchy and the sequential order of layout components within the document.

We jointly evaluate text and layout recognition through the mAP_{CER} [12]. It adapts the mean Average Precision (mAP) from object detection by using the Character Error Rate (CER) instead of Intersection over Union (IoU) for accuracy assessment. mAP_{CER} provides a nuanced measure of how well text regions are classified and recognized in a document. Since we are interested in offering a solution that works well in both perspectives, we monitor the mAP_{CER} metric to select the models for evaluation.

We also assess computational cost-related metrics: latency, throughput, and memory usage. Latency in this work refers to the time required for the model to process a single document. This metric is particularly crucial in applications where rapid processing of documents is necessary, such as real-time document analysis systems. Conversely, throughput measures the number of documents the model can process in a set time frame. This is especially important in scenarios involving bulk processing of documents, where efficiently handling a large volume of documents is a priority. Furthermore, we monitor the Graphics Processing Unit (GPU) memory usage.

4.2. Experimental Setup

Our models were developed using the PyTorch [32] framework with Automatic Mixed Precision (AMP) for enhanced efficiency. We use the xFormers tool [26] to implement our highly optimized attention building blocks. The computing environment comprised a GNU/Linux machine running the Linux Mint 21 system with an NVIDIA RTX 4070 TI GPU (12GB VRAM), an Intel Core i7-10700k 3.80 GHz, and 32 GB RAM. We employed the Adam optimizer [25] to optimize our models. The models underwent a pretraining phase on synthetic text lines using the CTC objective function [18], while the primary training on actual hand-

written document images utilized cross-entropy loss. An early stopping mechanism was implemented during pre-training, with a patience threshold set at 80 epochs. Our models were trained within a 40-hour window, selecting the best-performing model on the validation set for further testing. Throughout this research, we conduct experiments with DANCER and the state-of-the-art models DAN and Faster DAN for comparison purposes.

4.3. Training and Evaluation Results

Table 2 depicts the results on the single- and double-page READ dataset obtained by the DANCER and predecessor models after a 40-hour training period. We chose a training batch size that allowed us to fit all models within the GPU memory capacity, while the evaluation on the test set was made on a per-image basis. We used smaller versions of earlier models for double-page documents to fit into the GPU memory. Furthermore, we added an extra experiment where we selected the maximum batch size our model could reach using the available GPU, referred to as DANCER-Max.

The DANCER model can handle larger training batch sizes (6 for single pages and 3 for double pages) within the same 12GB memory constraint as the other models, effectively tripling the batch processing capacity. This capability indicates more efficient memory usage and can contribute to faster training times due to better GPU utilization. While DANCER reaches a maximum GPU usage of approximately 3.5GB processing single-page documents, the DAN variants consume around 9 GB, reducing the memory usage by roughly 61%. For double-page documents, the reduction reaches 65%. Besides the max memory peak, we can notice the difference in memory holds during the whole training, as shown in Figure 2. DANCER's low memory cost on double-page documents allows us to form mini-batches, while the same does not hold for reference works that can only be fed with single document images.

In single-page documents, DANCER's inference time is lower than other models, and it can achieve a 7× speedup in latency compared to DAN. Furthermore, DANCER can process a considerably higher number of single documents per minute than other models, reaching 118 manuscripts. As per Figure 3, we can also observe the throughput during training, revealing that DANCER-Max can handle approximately twice as many documents as Faster DAN. The ability to process more documents in less time might be crucial for large-scale document processing applications.

These improvements in the computational costs can be attributed to the use of octave convolutions and optimized attention operations. The low-frequency path of the octave convolution operates in a smaller resolution, reducing the convolution cost. Besides, the high computational cost caused by the quadratic nature of self-attention layers is clearly an issue when training transformers, and using memory-efficient attention and FlashAttention can significantly alleviate the cost since these operations optimize how attention scores and intermediate representations are stored and computed, minimizing redundant operations.

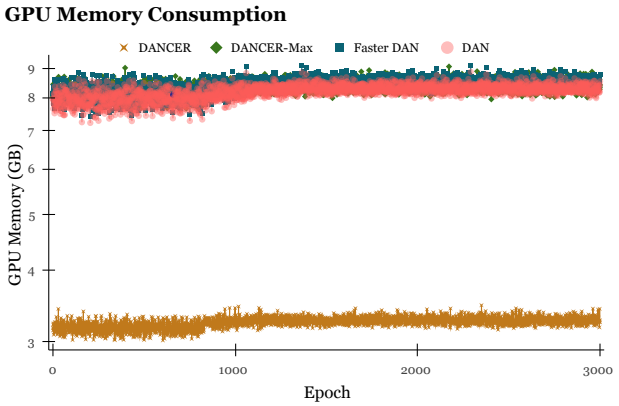


Figure 2. The graph presents DANCER's GPU memory consumption as a function of the training epochs on single-page document recognition.

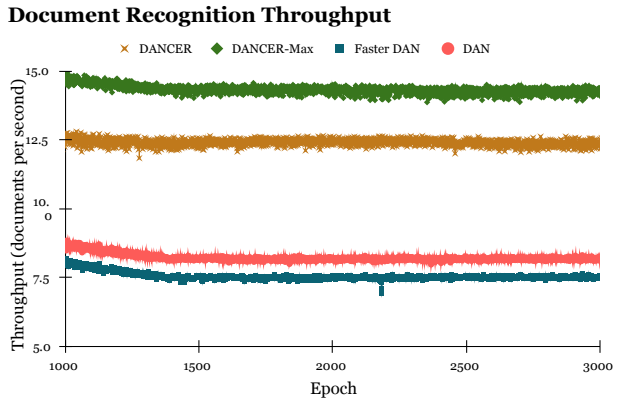


Figure 3. The graph presents the throughput in documents per second as a function of the training epochs computed on the validation set of the single-page READ dataset.

Regarding the recognition capabilities, the improvement in mAP_{CER} for double-page documents is especially noteworthy as this metric evaluates the joint recognition of both text and layout. Two-page manuscripts feature more complex layouts, with text flowing in parallel columns. This structure demands sophisticated mechanisms for accurately detecting page boundaries and determining the correct reading order. DANCER is the only model capable of reaching a mAP_{CER} over 95%, which marks a notable advancement over the predecessor models in correctly recognizing the text on the different text regions. Besides, DANCER achieves a WER of 13.34%, compared to DAN's 14.15%

Method	Training			Testing					
	Batch Size	Epochs	Max. GPU Mem.↓ (GB)	CER↓ (%)	WER↓ (%)	LOER ↓ (%)	mAP_{CER}↑ (%)	Lat.↓ (s)	Throughput↑ (docs/min)
Single-page recognition results									
DAN⊛	1	-	-	3.43	13.05	5.17	93.32	3.55	-
Faster DAN⊛	1	-	-	3.95	14.06	3.82	94.20	0.66	-
DAN	2	3113	8.9	3.63	14.36	4.61	**95.09**	3.55	16.9
Faster DAN	2	3282	9.1	3.53	13.84	4.39	94.56	0.66	90.7
DANCER	2	4821	**3.5**	**3.36**	13.73	**3.37**	94.73	**0.51**	**118.0**
DANCER-Max	6	5241	9.1	3.37	**13.00**	5.29	94.24	**0.51**	**118.0**
Double-page recognition results									
DAN⊛	1	-	-	3.70	14.15	4.98	93.09	8.50	-
Faster DAN⊛	1	-	-	3.88	14.97	**3.08**	94.54	1.90	-
DAN±	1	3436	9.0	4.31	15.46	4.03	92.89	4.73	12.7
Faster DAN±	1	3866	9.3	3.91	15.05	4.86	92.90	0.75	80.5
DANCER±	1	5243	**3.2**	3.98	14.58	4.39	93.67	**0.63**	**95.0**
DANCER	1	5178	3.3	3.64	14.37	4.51	94.44	0.87	68.9
DANCER-Max	3	5660	8.6	**3.37**	**13.34**	3.91	**95.08**	0.87	69.2

⊛ Official results extracted from the respective papers.
± These models had their number of decoder layers reduced from 8 to 5 layers to fit within GPU memory capacity.

Table 2. Results of the single-page and double-page experiments on the ICFHR 2016 READ dataset considering a time-based training time of 40 hours. ↓ indicates that lower is better while ↑ indicates that higher is better. We also included the official state-of-the-art reports obtained with 2-day pretraining and 4-day training. The throughput indicates the average number of processed documents per minute, and the latency refers to the average time in seconds that each model takes to process a single manuscript. The DANCER variants have approximately 6.93M learnable parameters, while DAN and Faster DAN have roughly 7.03M parameters.

in double-page documents, which suggests DANCER's refined ability to recognize words within complex layouts.

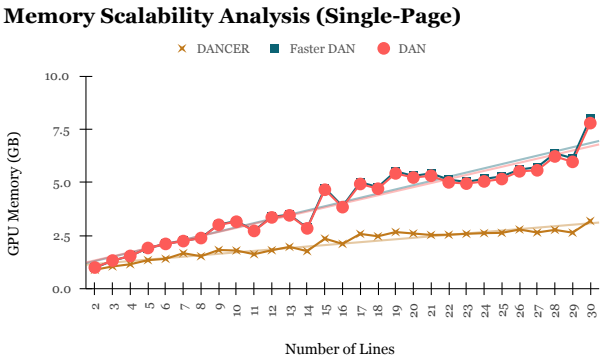
Overall, DANCER presented better recognition rates among the most computed metrics. Although one might argue that the improvements could be attributed to the training speedup allowing DANCER models to iterate over the training set more frequently, the fact that DANCER can achieve such small error rates under a 40-hour window being a shallower model compared to its predecessors (17 × 30 conv layers) is indicative of DANCER's effective feature learning ability with octave, depth-wise and gated convolutions.

4.3.1 Scalability Evaluations

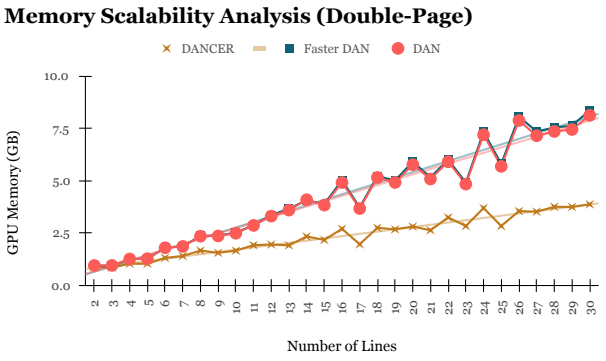
Figures 4a and 4b explore the memory scalability properties of DANCER using synthetic single-page and double-page documents to manipulate the number of lines fed to the model. We evaluate training DANCER with synthetic documents ranging from 1 to 30 text lines. This approach yields insights into how memory usage fluctuates as the number of text lines increases. The memory usage for both DAN and Faster DAN models increases roughly linearly with the increments in the number of lines. This is typical behavior as a larger input requires more memory to process. However, DANCER demonstrates a remarkably restrained increase in memory usage, suggesting an architecture less sensitive to the increase in the input size.

As expected, DANCER's design, encompassing octave convolution and efficient attention techniques, leads to a solution with better memory scalability properties. The fact that DANCER is less affected by document length makes our solution potentially more robust and suitable to tackle the handwritten document recognition task, which typically involves full-page document images.

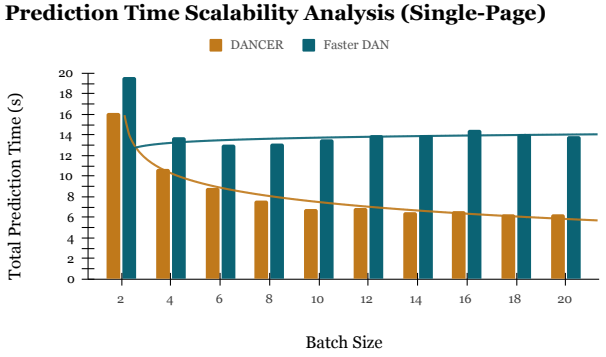
We conduct a second scalability analysis evaluating how the total prediction time on the test set evolves as we increase the batch size. As shown in Figure 4c, increasing the batch size substantially decreases DANCER's prediction time. In contrast, Faster DAN's prediction time remains relatively constant or decreases only slightly. The difference in behaviors could be explained by the parallelism exploration of DANCER's memory-efficient attention and FlashAttention. These optimized blocks are designed to exploit the parallel processing of modern GPU



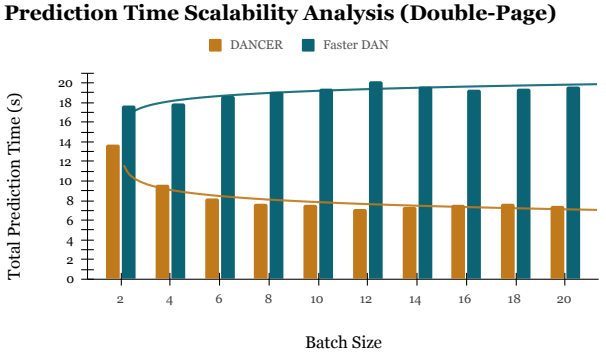
(a) Memory use on the single-page dataset training as a function of the number of lines in the synthetic samples.



(b) Memory use on the double-page dataset training as a function of the number of lines in the synthetic samples.



(c) Total prediction time on the single-page dataset inference at the test set as a function of the batch size.



(d) Total prediction time on the double-page dataset inference at the test set as a function of the batch size.

Figure 4. Analysis of the scalability considering the GPU memory usage during training and the latency in the inference.

more effectively than standard attention algorithms, which becomes evident in our analysis as we increase the number of documents simultaneously processed. DANCER's improvements throughout different batch sizes convey that the model's architecture is robust across different workloads.

Figure 4d depicts the prediction time on double-page documents. In this case, the difference in performance between the Faster DAN and DANCER is even more pronounced, with Faster DAN taking more than double the time DANCER takes to complete the test set evaluation. As double-page documents have a higher spatial resolution, the impact of downsampling in the low-frequency path of the octave layers becomes more evident. This highlights DANCER's efficient design in processing complex documents, resulting in a faster and more accurate solution. It can offer a more viable solution for practical deployment cases where many documents need to be processed in batches, such as digitizing entire archives. Overall, the findings illustrate our model's enhanced scalability for HDR, effectively managing computational costs in high-demand scenarios.

5. Conclusion

This research introduces DANCER, a computationally efficient document attention model for handwritten document recognition. DANCER's design incorporates highly optimized building blocks that can bring computing optimizations while maintaining strong recognition accuracy. We conducted experiments using the challenging historical dataset READ in both single- and double-page setups. The results position DANCER as a memory-efficient and faster alternative to previous state-of-the-art solutions. The enhancements that DANCER brings in terms of memory efficiency, speed, and accuracy suggest its potential to revolutionize practical applications, especially in situations with limited resources or where rapid processing of large volumes of documents is required.

Beyond advancing our understanding of HDR, this paper also establishes a foundation for future exploration within this promising domain. In future work, we aim to evaluate DANCER across more datasets further to extend our knowledge of its capabilities in diverse contexts.

References

[1] Killian Barrere, Yann Soullard, Aurélie Lemaitre, and Bertrand Coüasnon. A light transformer-based architecture for handwritten text recognition. In *Int. Work. on Document Analysis Systems*, pages 275–290. Springer, 2022. 2

[2] Théodore Bluche. Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. *Advances in neural information processing systems*, 29, 2016. 2

[3] Théodore Bluche and Ronaldo Messina. Gated convolutional recurrent neural networks for multilingual handwriting recognition. In *14th Int conference on document analysis and recognition (ICDAR)*, pages 646–651. IEEE, 2017. 2

[4] Théodore Bluche, Jérôome Louradour, and Ronaldo Messina. Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention. In *14th Int. conference on document analysis and recognition (ICDAR)*, pages 1050–1055. IEEE, 2017. 2

[5] Manuel Carbonell, Joan Mas, Mauricio Villegas, Alicia Fornés, and Josep Lladós. End-to-end handwritten text detection and transcription in full pages. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, pages 29–34. IEEE, 2019. 3

[6] Manuel Carbonell, Alicia Fornés, Mauricio Villegas, and Josep Lladós. A neural model for text localization, transcription and named entity recognition in full pages. *Pattern Recognition Letters*, 136:219–227, 2020. 3

[7] Silvia Cascianelli, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Boosting modern and historical handwritten text recognition with deformable convolutions. *International Journal on Document Analysis and Recognition (IJDAR)*, pages 1–11, 2022. 2

[8] Dayvid Castro, Byron LD Bezerra, and Meuser Valenca. Boosting the deep multidimensional long-short-term memory network for handwritten recognition systems. In *2018 16th international conference on frontiers in handwriting recognition (ICFHR)*, pages 127–132. IEEE, 2018. 2

[9] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3435–3444, 2019. 2

[10] Denis Coquenet, Clément Chatelain, and Thierry Paquet. Span: a simple predict & align network for handwritten paragraph recognition. In *International Conference on Document Analysis and Recognition*, pages 70–84. Springer, 2021. 1, 2

[11] Denis Coquenet, Clément Chatelain, and Thierry Paquet. End-to-end handwritten paragraph text recognition using a vertical attention network. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 45(1):508–524, 2022. 1, 2, 4

[12] Denis Coquenet, Clément Chatelain, and Thierry Paquet. Dan: a segmentation-free document attention network for handwritten document recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2023. 1, 3, 5

[13] Denis Coquenet, Clément Chatelain, and Thierry Paquet. Faster dan: Multi-target queries with document positional encoding for end-to-end handwritten document recognition. *arXiv preprint arXiv:2301.10593*, 2023. 1, 3, 4, 5

[14] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022. 2, 3, 5

[15] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017. 4

[16] Arthur Flor de Sousa Neto, Byron Leite Dantas Bezerra, Alejandro Héctor Toselli, and Estanislau Baptista Lima. Htr-flor: A deep learning system for offline handwritten text recognition. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 54–61. IEEE, 2020. 2

[17] Quentin Fournier, Gaétan Marceau Caron, and Daniel Aloise. A practical survey on faster and lighter transformers. *ACM Computing Surveys*, 55(14s):1–40, 2023. 1

[18] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006. 5

[19] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[21] R Reeve Ingle, Yasuhisa Fujii, Thomas Deselaers, Jonathan Baccash, and Ashok C Popat. A scalable handwritten text recognition system. In *2019 Int. Conf. on Document Analysis and Recognition (ICDAR)*, pages 17–24. IEEE, 2019. 2

[22] José Carlos Aradillas Jaramillo, Juan José Murillo-Fuentes, and Pablo M Olmos. Boosting handwriting text recognition in small databases with transfer learning. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 429–434. IEEE, 2018. 2

[23] Lei Kang, Pau Riba, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. Pay attention to what you read: non-recurrent handwritten text-line recognition. *Pattern Recognition*, 129:108766, 2022. 2

[24] Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. On the computational complexity of self-attention. In *International Conference on Algorithmic Learning Theory*, pages 597–619. PMLR, 2023. 3

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Preprint arXiv:1412.6980*, 2014. 5

[26] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022. 5

[27] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, pages 707–710. Soviet Union, 1966. 5

[28] Johannes Michael, Roger Labahn, Tobias Grüning, and Jochen Zöllner. Evaluating sequence-to-sequence models for handwritten text recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1286–1293. IEEE, 2019. 1, 2

[29] Samah Mohammed and Nick Teslya. Joint recognition of text and layout in historical russian documents. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 23:585–594, 2023. 3

[30] Bastien Moysset and Ronaldo Messina. Are 2d-lstm really dead for offline text recognition? *Int. Journal on Document Analysis and Recognition (IJDAR)*, 22(3):193–208, 2019. 2

[31] Bastien Moysset, Christopher Kermorvant, and Christian Wolf. Full-page text recognition: Learning where to start and when to stop. In *14th Int. Conference on Document Analysis and Recognition (ICDAR)*, pages 871–876. IEEE, 2017. 3

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[33] Jason Poulos and Rafael Valle. Character-based handwritten text transcription with attention networks. *Neural Computing and Applications*, pages 1–11, 2021. 2

[34] Joan Puigcerver. Are multidimensional recurrent layers really necessary for handwritten text recognition? In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 67–72. IEEE, 2017. 1, 2

[35] Markus N Rabe and Charles Staats. Self-attention does not need $o(n^2)$ memory. *ArXiv:2112.05682*, 2021. 2, 3, 5

[36] Joan Andreu Sanchez, Veronica Romero, Alejandro H Toselli, and Enrique Vidal. Icfhr2016 competition on handwritten text recognition on the read dataset. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 630–635. IEEE, 2016. 5

[37] Martin Schall, Marc-Peter Schambach, and Matthias O Franz. Multi-dimensional connectionist classification: Reading text in one step. In *13th Int. Workshop on Document Analysis Systems (DAS)*, pages 405–410. IEEE, 2018. 2

[38] Annapurna Sharma and Dinesh Babu Jayagopi. Towards efficient unconstrained handwriting recognition using dilated temporal convolution network. *Expert Systems with Applications*, 164:114004, 2021. 2

[39] Sumeet S Singh and Sergey Karayev. Full page handwriting recognition via image to sequence extraction. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part III 16*, pages 55–69. Springer, 2021. 3

[40] Wanchen Sui, Qing Zhang, Jun Yang, and Wei Chu. A novel integrated framework for learning both text detection and recognition. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2233–2238. IEEE, 2018. 3

[41] Vasiliki Tassopoulou, George Retsinas, and Petros Maragos. Enhancing handwritten text recognition with n-gram sequence decomposition and multitask learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10555–10560. IEEE, 2021. 2

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[43] Paul Voigtlaender, Patrick Doetsch, and Hermann Ney. Handwriting recognition with large multidimensional long short-term memory recurrent neural networks. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 228–233. IEEE, 2016. 1, 2

[44] Christoph Wick, Jochen Zöllner, and Tobias Grüning. Rescoring sequence-to-sequence models for text line recognition with ctc-prefixes. In *International Workshop on Document Analysis Systems*, pages 260–274. Springer, 2022. 2

[45] Curtis Wigington, Chris Tensmeyer, Brian Davis, William Barrett, Brian Price, and Scott Cohen. Start, follow, read: End-to-end full-page handwriting recognition. In *Proceedings of the European conference on computer vision (ECCV)*, pages 367–383, 2018. 3

[46] Mohamed Yousef and Tom E Bishop. Origaminet: weakly-supervised, segmentation-free, one-step, full page text recognition by learning to unfold. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14710–14719, 2020. 3