Enhancing CRNN HTR Architectures with Transformer Blocks

George Retsinas[1], Konstantina Nikolaidou[2], and Giorgos Sfikas[3,4]

[1] National Technical University of Athens, Greece
gretsinas@central.ntua.gr
[2] Luleå University of Technology, Sweden
konstantina.nikolaidou@ltu.se
[3] University of West Attica, Greece
gsfikas@uniwa.gr
[4] University of Ioannina, Greece

Abstract. Handwritten Text Recognition (HTR) is a challenging problem that plays an essential role in digitizing and interpreting diverse handwritten documents. While traditional approaches primarily utilize CNN-RNN (CRNN) architectures, recent advancements based on Transformer architectures have demonstrated impressive results in HTR. However, these Transformer-based systems often involve high-parameter configurations and rely extensively on synthetic data. Moreover, they lack focus on efficiently integrating the ability of Transformer modules to grasp contextual relationships within the data. In this paper, we explore a lightweight integration of Transformer modules into existing CRNN frameworks to address the complexities of HTR, aiming to enhance the context of the sequential nature of the task. We present a hybrid CNN image encoder with intermediate MobileViT blocks that effectively combines the different components in a resource-efficient manner. Through extensive experiments and ablation studies, we refine the integration of these modules and demonstrate that our proposed model enhances HTR performance. Our results on the line-level IAM and RIMES datasets suggest that our proposed method achieves competitive performance with significantly fewer parameters and without integrating synthetic data compared to existing systems.

Keywords: Handwriting Text Recognition · Transformer Modules

1 Introduction

Handwritten Text Recognition (HTR) systems aim to enable machines to recognize human writing. HTR is utilized in several applications such as the digitization of handwritten and historical documents for easier storage, preservation, and analysis. The task poses several challenges due to the unique and varying characteristics and calligraphic style between different writers as well as within each writer. The complexity of the task requires multimodal approaches that

bridge Computer Vision and Natural Language Processing techniques to understand and interpret human handwriting.

Research in learning for HTR has been centered around formulating a suitable model that can capture faithfully the intrinsic traits of the process of handwriting. Perhaps the most salient feature of handwriting is its sequential nature, and before the advent of Deep Learning most successful models have been using variants of Hidden Markov Modeling [4]. Recurrent Neural Networks have been made the deep model of choice for HTR, and they have started gaining prominence especially after a number of key advances, such as the introduction of variants that can deal with vanishing/exploding gradients (Long Short Term Memory, Gated Recurrent Units) and the introduction of Connectionist Temporal Classification (CTC)-based objectives [12]. Difficulties in training Recurrent-only architectures have incentivized researchers to employ convolutional components, or whole convolutional backbones and integrate them with a Recurrent model. Standard architectures in this vein consist of an encoder-decoder framework, by deploying a Convolutional Neural Network (CNN) as a backbone to create the image features and a Recurrent Neural Network (RNN) for the sequential modeling part.

In the past few years, Transformers [30] have dominated the field of Natural Language Processing due to the powerful context created by the self-attention mechanism. Transformers enjoy a global receptive field, as opposed to recurrent modeling, which enables them to capture long-range dependencies. Combined with the positional embedding of tokens, this enables an accurate treatment of sequential data as well as provides a much richer model than recurrent networks. The mechanism has been explored on visual tasks with Vision Transformers (ViT) by treating the images as sequence patches instead of text tokens. ViT have demonstrated promising results within image data, suggesting potential benefits for enhancing HTR performance. However, this increased effectiveness does not come without a price: Transformer-based architectures are known to be exceptionally data-hungry in terms of the sizes of the training set size, as well as the large size of the model (e.g., [18]).

In light of the above considerations, we believe that a careful analysis and exploration regarding how to integrate self-attention models in a non-Transformer pipeline optimally should be considered. Our contributions can be summarized as follows:

1. We explore the use of Transformer modules in both the visual and textual parts of the HTR system. Unlike most previous work, special emphasis is given to a (relatively) resource-constrained setting.
2. We propose a hybrid CNN-MobileViT image encoder to model the sequences of the images.
3. We present a thorough ablation study on the Transformer parameters in both the encoding and decoding parts and compare it with other State-of-the-Art HTR methods.

2 Related Work

Handwritten text is commonly processed sequentially by methods that learn to model the input by handling sequences of characters in various lengths. Motivated by formulating the sequential nature of text in terms of a model with a suitable inductive bias, earlier methods were based on Hidden Markov Models (HMM) [9,11,4]. Textual latent features were defined to form a Markov chain, with each component related to an observed column of the text image. After the revolution brought by Deep Learning in the mid-2010s, HMMs have been steadily superseded by models based on Recurrent Neural Networks [13].

Advances in Handwritten Text Recognition. When coupled with Connectionist Temporal Classification (CTC)-based objectives, which enable loss computation during training without requiring exact alignment between prediction and target, RNNs have served as the foundation for numerous excellent-performing systems [12,24]. Alternatively, Sequence-to-Sequence models offer a different approach to HTR, separating encoding into a feature vector and decoding into the target string as distinct network components [27]. In comparison to Convolutional Neural Networks (CNNs), RNNs possess the advantage of capturing sequential information dependencies, often resulting in superior HTR models. Encoding prior knowledge about input characteristics is primarily achieved through the inherent biases within each model. Convolutional layers, for instance, model statistical dependencies within each time frame or spatial dependencies of line image cues with respect to nearby pixels. Conversely, recurrent layers encode the belief that data are inherently sequential, with bidirectional recurrent variants adept at capturing forward and backward dependencies. However, RNNs are recognized for being challenging to train and converge to satisfactory solutions. Thus, architectures combining convolutional and recurrent components have been proposed [27,25,20,28]. A prevalent approach involves a convolutional backbone transforming input segmented images into a useful feature map, which is subsequently pooled or reshaped into a sequence of features fed into a recurrent component. For convolutional-recurrent model architecture, incorporating an auxiliary CTC-based component into the primary recurrent network has proven effective, serving as a penalty for cross-entropy loss over a recurrent decoder. In practice, employing a fully convolutional CTC shortcut encompassing 2D and 1D convolutions translates to fewer recurrent components and faster convergence. The work presented in [25] proposes best practices in standard CNN-RNN HTR systems trained with CTC loss and manages to achieve state-of-the-art results without the use of synthetic data.

Transformers. The advent of Transformers [30] and their ability to handle long-range dependencies has introduced an alternative to the sequential RNN processing with the use of the attention mechanism. Initially introduced within the realm of Natural Language Processing [30], they have been adapted for tasks amenable to sequential processing. At the heart of Transformers lies the self-attention mechanism, wherein input sequence vectors undergo transformation into "keys," "queries," and "values" via learnable, shared operations, effectively creating a feature dictionary. These features are then combined through

a softmax-weighted average and further processed via a fully-connected layer to yield an output sequence. Experimentally, employing multiple transformations on the same inputs, known as the multi-head variant of self-attention, has proven effective. These multi-head self-attention operations can be organized into stacks to form transformer layers. Recent endeavors have explored the application of Transformers in Document Image Processing applications, including HTR, invariably showcasing excellent results [3,36,18].

In recent years, a notable number of works have been done on using Transformers in HTR. [8] have replaced Recurrent components with Self-attention blocks, showing that this leads to a clear advantage in terms of training time and model complexity requirements. A Sequence-to-Sequence Transformer is proposed in [1], and experiments are carried out with an architecture that is much more compact than the norm in Transformer models [2]. In [16], they combine Transformer blocks with convolutional components for the visual encoding part and recurrent components are replaced by a Transformer decoder that attends to both the textual and visual part. An attention-based sequence-to-sequence system that consists of a CNN-RNN feature extractor and a separate RNN architecture for the character sequence decoding, placing an attention mechanism between the visual encoding and textual decoding part, is presented in [23]. TrOCR [18] is an end-to-end text recognition system consisting of image and text Transformer models pre-trained on large-scale synthetic data. DTrOCR [10] explores using a decoder-only Transformer architecture, taking advantage of a generative language model (Generative Pretrained Transformer, GPT) that is pretrained on a large corpus. Unlike this line of works, our approach focuses on exploring the power of Transformers without synthetic pre-training.

Despite these successes, Transformers in HTR are associated with a number of disadvantages: large model size and a large training dataset –often synthetic in practice– are among these. [32] urge further caution, as they note that they can be related to drawbacks such as struggling to handle text repetitions. Models reliant solely on Transformer structures tend to exhibit significantly larger sizes compared to their non-transformer counterparts [33]. Also, they are notably even more data-hungry than what is the (already high) norm in Deep Learning, often requiring large-scale synthetic datasets to unravel their full potential. With the current work, we aim to address these issues by exploring an optimal trade-off with respect to coupling visual transformer blocks with convolutional and recurrent components and leveraging advances such as the CTC shortcut [25]. We show that the proposed architecture leads to State-of-the-Art results without requiring the use of synthetic pre-training like previous work [8,18,10].

3 Proposed System

The motivation of the proposed system is straightforward and can be condensed into this question: “Can we utilize the effectiveness of attention in lightweight architectures?”. Several aspects are crucial to address the aforementioned question:

– First, we have to consider how the innate large transformer architectures can be integrated into lightweight systems. To this end, we considered a hybrid CNN-Transformer architecture, motivated by the MobileViT model [22].
– Second, we are interested in a setting of limited data, namely, using the standalone training set of datasets without any external/synthetic data augmentation. This can be a notable issue with transformers, which are proven to be very effective when a very large amount of training data is used [2,18,10]. While such setup is not directly useful in practice for the considered languages / datasets, it is an important point in low-resources languages.
– Lastly, we aim to highlight optimization tweaks that can be utilized to support the training of such hybrid architectures under the aforementioned setting of (relatively) limited data.

3.1 Hybrid HTR Architecture

The design of our architecture aims to combine the benefits of both convolutional and transformer architectures for feature extraction, namely a lightweight and easily trainable model, as in typical CNNs, and a holistic ability to integrate context, as in transformers. To this end, we followed the success of MobileViT [22] and used the same transformer components, referred to as MobileViT blocks. To be in line with the handwritten text recognition task, we opted for a task-customized CNN backbone, as in [25]. The transformed blocks were then intervened between typical convolutional blocks, as in [22]. The aforementioned hybrid architecture is the backbone of our system, responsible for the feature extraction step. The extracted feature map should be translated to character probabilities in order to perform the text recognition step. This is performed through the head module. Following [25], a recurrent head is selected, while also a transformer head was considered as an alternative. Both head variants aim to capture context and assist character prediction. The system is trained using the Connectionist Temporal Classification (CTC) loss [12]. Figure 1 depicts the proposed architecture and its components. A description of these components will be provided in detail in the following subsections.

Hybrid Backbone Our system utilizes a hybrid CNN-Transformer backbone, where the CNN part is the same as in [25] and the transformer modules are MobileViT blocks, as defined in [22]. In more detail, the backbone takes a text image as input and outputs a sequence of features to be processed by the head of the model. It consists of:

– A first layer of a single 7×7 convolution and 2×2 stride to downscale the initial feature map.
– Groups of sequential 3×3 *ResNet blocks* [15]. There are three such groups in total. The first has 2 blocks, while the latter two have 4.
– Between these groups of blocks a 2×2 *max-pooling* of stride 2 is used to further downscale the produced feature map.

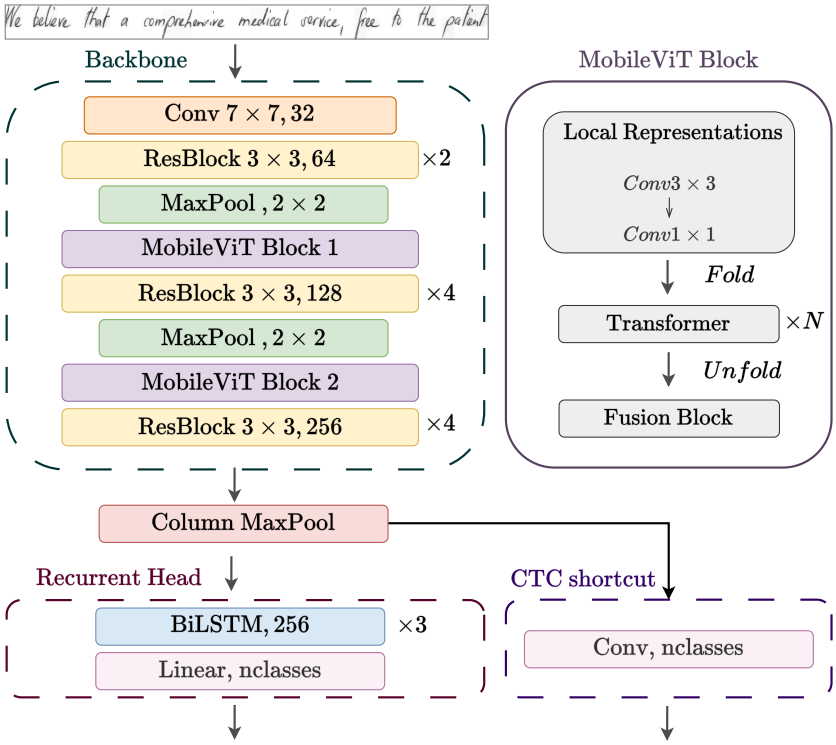

Fig. 1. Overview of the model architecture. We present the hybrid CNN backbone with the intermediate MobileViT blocks and the Recurrent head with the CTC shortcut. The MobileViT block components are depicted on the right upper part.

- The *MobileViT blocks* are added immediately after the max-pooling operation. This means that we considered 2 MobileViT in total to be added to our architecture, as seen in Fig. 1.
- A *column-wise max-pooling* operation flattens the 3D visual feature map, which is the output of the backbone, into a sequence of features.

MobileViT blocks are the core component of the proposed architectural enhancement. To this end, let us describe their functionality in detail. A high-level description of the MobileViT structure is depicted in Fig. 1, where we can distinguish three sub-blocks:

- The *Local Representation* block which simply transforms the input via a set of convolutional layers. It prepares the input by calculating local correlations before the upcoming patch-based transformer step.
- The Transformer block uses a number of stacked transformation modules consisting of Multi-Head (self-)Attention, fully connected, and normalization layers in a residual manner. Before this block, a folding operation into

patches is used. Specifically, given a patch size of $p \times p$, the block's input of size $\mathbb{R}^{H \times W \times d}$ is transformed into a tensor of size $\mathbb{R}^{P \times D \times d}$, where $P = HW/p^2$ and $D = p^2$. The inverse operation of unfolding is applied after the transformer block to return to a spatial tensor formulation. This step provides the necessary holistic information to learn long-term spatial dependencies.
- The final *Fusion Block* concatenates the input of the whole MobileViT block and the output of the previous transformer block and processes it through a final convolutional layer that combines the local and the global information learned by the whole MobileViT block.

The appeal of MobileViT blocks is two-fold. First, they introduce spatial inductive bias as the authors highlight [22], which seems to be an issue for ViT optimization [34]. Furthermore, they provide a lightweight approach to generating context-aware global embeddings, having a receptive field of the whole spatial domain without losing the order of the patches and without the need for positional encoding.

The intuition of using such transformer blocks in the context of handwritten text recognition can be summarized as the ability to correlate intermediate features along the whole image, hinting towards an implicit writing style adaptation. In other words, by having this global receptive field step, we can understand and consequently adapt to different writing styles, which may not be detectable by the local intermediate convolutional layers in the default setting.

Head Variants We explore two types of heads for the decoding part that generates the character sequences: a recurrent and a transformer head.

- *Recurrent Head:* Similar to [25], our recurrent head relies on a stack of three Bidirectional Long Short-Term Memory (BiLSTM) units with a hidden size of 256, which is projected linearly to the number of character classes.
- *Transformer Head:* We replace the recurrent head with a transformer head that operates directly on the sequences and aim to explore the ability of a self-attention architecture to capture context. Specifically, we use only the encoder part of the standard Transformer [30], which is comprised of N stacked blocks of Multi-Head Attention layers. To precisely capture the order of the sequential input, we also employ a positional encoding layer.

Both variants convert the input sequence of feature dimension d to a sequence of the same length with feature size $n_{classes}$, i.e. the number of possible character tokens, including the blank character required by CTC. Thus, training is performed via CTC, and evaluation is performed via greedy decoding, namely by choosing the character with the highest probability at each step, reducing consecutive identical characters into a single one, and then removing the blank tokens.

An important distinction here is that the Transformer head is designed for the CTC loss. Thus, we essentially rely on the self-attention concept to capture

long-term dependencies, along with the required positional encoding. A different approach would be to use the complete encoder-decoder architecture of the typical Transformer, as used in NLP task [30]. This assumes a sequence-to-sequence rationale and, thus, a different loss, where character predictions are performed sequentially (training can be parallelized). To avoid such computationally intense architectures, we opted for the more lightweight solution discussed earlier.

Finally, we also use an additional convolutional-only head as an auxiliary shortcut branch to assist optimization, as done in [25]. Specifically, this auxiliary head consists of a single 1D convolution layer that outputs features of dimensions equal to the possible character tokens ($n_{classes}$). It is also trained using CTC loss in parallel to the main branch. Essentially, a multi-task loss is used by adding the two individual CTC losses with the appropriate weights, namely the main as it is and the auxiliary scaled down by 0.1. The concept of this shortcut path is to assist the training by providing a straightforward path for the backward propagation, akin to the residual rationale, providing high-quality encodings at the output of the backbone to be further processed by the context-aware head.

3.2 Optimization

As stated before, Transformer modules can lead to subpar performance if the amount of training data is limited. Considering common HTR settings, without the use of external data (e.g., synthetic), the training set typically has a few thousand lines (e.g., IAM has around $6K$ lines for training). Despite the typical augmentation step of affine transformations (see Experiments section for details) during training, such amount of data could prevent our system from unlocking its potential, even if only two MobileViT blocks are considered.

End-to-end training of the proposed system on limited data may underperform [17]. To this end, we considered an alternative training scheme, where two distinct training steps are considered. First, we train the architecture without the transformer blocks. Essentially, we train the same architecture as in [25]. Then, we use this pretrained model to initialize all non-transformer weights and re-train the whole architecture. This simple workaround could potentially further "unlock" the effectiveness of the transformer blocks by providing a good initialization of the whole system. In essence, we force the transformer to learn well-performing context-aware representations under this specific framework.

4 Experiments

4.1 Experimental Setup

Datasets & Metrics: We evaluate our proposed system on the IAM database [21] on the line-level setting. We present a thorough ablation study on the Transformer parameters of the backbone and head of the system. We further present an ablation study on the different components of the system in the backbone and the head. The ablation experiments are performed on IAM, using the writer

independent train/validation/test split, similar to [24,25]. The reported results present the Character Error Rate (CER) and Word Error Rate (WER) metrics. Given the results of the ablation study, we compare the performance of our best parameter combination model with other state-of-the-art systems on the IAM and RIMES [14] test sets while taking into account the presence of synthetic data and the number of parameters.

Data Pre-processing: The pre-processing steps follow the practices presented in [25]. Specifically:

- Images are fixed to a size of 128×1024. The aspect ratio is kept if the image is smaller than the aforementioned resolution, and a padding operation is considered. Otherwise, the image is resized to the required size.
- During training, spatial affine transformation, and Gaussian noise are used for on-the-fly data augmentation.
- Target text is extended by adding the space token before and after the actual text in order to correspond to the spaces created by the padding operation on the images.

Training Details: The proposed HTR system is trained for 240 epochs with an initial learning rate of 0.001 that decreases by a factor of 0.1 at epochs 120 and 180, following the training scheme of [25]. In our two-step training scheme, we initialize the pre-trained layers with a learning rate of 0.0001. This choice is made to ensure training stability and preservation of the knowledge captured in the layers for effective transfer to the second training step. We use AdamW [19] as the optimizer and a batch size of 16 samples. The implementation is based on the PyTorch framework, and every experiment runs on a single A100 GPU.

4.2 Ablation Study

We explore the values of the different parameters present in the Transformer modules utilized in the backbone or head to obtain a final setting with good performance and a reasonable number of parameters. To this end, we report the CER and WER on the validation set of IAM without the use of the pre-trained layers proposed in our pre-training scheme.

The Transformer architecture consists of several key parameters that impact its efficiency and performance. The model dimension d_{model} corresponds to the size of the embeddings used in the model and the number of layers N to the number of identical stacked Transformer blocks. The self-attention in the multi-headed attention layer is performed in parallel in h number of heads, where every head has a dimension d_{head}. Finally, the feed-forward network dimension mlp_{dim} dictates the size of the inner fully connected layer of the Transformer that appears after the Multi-headed attention layer.

We explore several combinations of the model dimension d_{model}, the number of layers N, the number of heads h in the multi-headed attention, and the head dimension d_{head} in the Transformer module of the two MobileViT blocks. In every experiment of the ablation study, the feed-forward network dimension is

Table 1. Ablation on the d_{model} of the transformer present in the MobileViT blocks for $N = 1$ layer and $h = 1$ heads. The head dimension in every case is $d_{head} = d_{model}/h$.

$N = 1$, $h = 1$			
d_{model}	CER(%)↓	WER(%)↓	# params
64	3.14 ± 0.02	11.54 ± 0.31	10.86M
80	3.07 ± 0.08	11.27 ± 0.24	10.90M
128	3.19 ± 0.04	11.68 ± 0.21	11.08M
256	3.48 ± 0.08	12.48 ± 0.11	11.92M

Table 2. Ablation on the number of heads h of the transformer present in the MobileViT blocks for $N = 1$ layer and $d_{model} = 80$ heads. The head dimension in every case is $d_{head} = d_{model}/h$.

$N = 1$, $d_{model} = 80$					
h	d_{head}	d_{model}	CER(%)↓	WER(%)↓	# params
1	80	80	3.07 ± 0.08	11.27 ± 0.24	10.90M
2	40	80	3.14 ± 0.11	11.57 ± 0.38	10.90M
4	20	80	3.13 ± 0.10	11.45 ± 0.34	10.90M
8	10	80	3.00 ± 0.06	11.02 ± 0.34	10.90M

set to $mlp_{dim} = 2 * d_{model}$, and we keep a fixed patch size dimension of $(4, 4)$ and $(8, 8)$ in the first and second MobileViT block, respectively.

We begin the exploration using $N = 1$ layer and $h = 1$ and experiment with model dimension d_{model} for values 64, 80, 128, and 256. In this scenario, the combination of parameters is the same for both blocks, and the head dimension is set as $d_{head} = d_{model}/h$. The results of the ablation, along with the number of parameters, are presented in Table 1. The results indicate that a larger value of the model dimension decreases the performance. In comparison, a model dimension of 80, which is closer to the smaller value we picked, gives the best result. Given the results of Table 1, we proceed with $d_{model} = 80$ and gradually increase the number of heads. Table 2 shows that for a constant number of model parameters, the highest number of heads, which is 8, achieves the best performance. Following the best-performing setting from the previous ablation again, we continue to explore how the model performs if we keep a fixed $d_{head} = 80$ and increase the model dimension along with the heads as $d_{model} = h * d_{head}$. The results are presented in 3, where we can observe that the smaller the model, the better the performance. To conclude with the setting choice, we use the best-performing combination, which is $d_{model} = 80$, $h = 8$, and $d_{head} = 10$ and increases the Transformer layers N. The results of the layer exploration are presented in Table 5. Similar to the previous ablation steps, the best performance on the validation set is achieved by the smaller model, making us proceed with $N = 1$ for our proposed system.

We replace the BiLSTM head with a standard Transformer block while using the CNN head without the MobileViT blocks and explore a few combinations

Table 3. Ablation on the increase of heads h and model dimension d_{model} of the transformer present in the MobileViT blocks for $N = 1$ layer and $d_{head} = 80$ heads.

$N = 1$, $d_{head} = 80$					
h	d_{head}	d_{model}	CER(%)↓	WER(%)↓	# params
1	80	80	3.07 ± 0.08	11.27 ± 0.24	10.90M
2	80	160	3.22 ± 0.19	11.89 ± 0.57	11.24M
4	80	320	3.36 ± 0.16	12.26 ± 0.23	12.53M

of the head's parameter values. The results are presented in Table 6. One can observe that replacing the BiLSTM head with a Transformer head that operates directly on the output sequences is not straightforward enough to show an improved performance and requires further investigation.

Table 4. Ablation on increasing the d_{model} of the transformer present in the second MobileViT block for $N = 1$ layer $d_{model} = 80$ on the first MobileViT block.

$N = 1$					
block	d_{model}	h	CER(%)↓	WER(%)↓	# params
1	80	8	3.00 ± 0.06	11.02 ± 0.34	10.90M
2	80	8			
1	80	8	3.14 ± 0.08	11.47 ± 0.17	11.07M
2	160	8			
1	80	8	3.04 ± 0.05	11.15 ± 0.10	11.07M
2	160	16			

Finally, we explore the impact of our proposed MobileViT blocks by conducting the experiment with and without their addition. We present the results on both validation and test sets in Table 7.

4.3 Comparison with State-of-the-Art

We evaluate our proposed method, utilizing the optimal setting identified through our ablation study, and compare it with leading state-of-the-art systems of various types of line-level recognition on IAM and RIMES test sets. As demonstrated in Table 8, our proposed system attains a CER of 4.22% and a WER of 14.58% on the IAM test set. This performance is achieved using the pretraining scheme mentioned in 3.2 and without the use of synthetic data augmentation. Notably, our CER score matches that of the $\text{TrOCR}_{\text{SMALL}}$ [18], yet our model requires significantly fewer parameters - even without training on extra data, underscoring the effectiveness of our integration strategy. On the RIMES dataset, our method achieves a CER of 2.70% and a WER of 9.46%, placing it among the top-performing methods, but leaving room for improvement. It is important to

Table 5. Ablation on the number of layers N for $d_{model} = 80$ and $h = 8$ in the transformer module of both MobileViT blocks.

N	d_{model}	h	CER(%)↓	WER(%)↓	# params
1	80	8	3.00 ± 0.06	11.02 ± 0.34	10.90M
2	80	8	3.13 ± 0.09	11.53 ± 0.25	11.00M
3	80	8	3.06 ± 0.11	11.33 ± 0.42	11.11M

Table 6. Ablation on the Transformer head without intermediate ViT blocks in the backbone.

$\mathbf{d_{model}}$	**layers**	**heads**	$\mathbf{d_{head}}$	**CER(%)↓**	**WER(%)↓**
32	1	1	32	5.90 ± 0.04	20.39 ± 0.20
32	1	2	32	5.89 ± 0.04	20.39 ± 0.13
64	1	2	32	5.99 ± 0.17	20.64 ± 0.51
80	1	2	40	6.07 ± 0.22	20.93 ± 0.61
256	1	1	256	6.10 ± 0.19	21.06 ± 0.60

highlight that while the lowest CER is achieved by Diaz et al.[6], DTrOCR[10] and TrOCR [18] models, our model operates with notably fewer parameters. This is a vital factor, considering the resource constraints that typically appear in practical applications. Moreover, these three models not only use considerably more data for training, but they also augment theirs results with Language Models (LMs). Diaz et al.[6] use a CTC decoding approach equipped with an external LM, while DTrOCR[10] and TrOCR [18] decode word tokens and not characters. On the other hand, Coquenet et al. [5] has a very lightweight approach, with some similar architecture elements, that achieves vert good results in both IAM and RIMES datasets further validating our main point on the importance of properly designing an HTR system.

Our results suggest that the integration of the MobileViT blocks into a conventional CRNN architecture not only retains but can enhance the performance of the HTR task. This hybrid approach offers a compelling alternative to systems that heavily depend on synthetic data, achieving competitive performance scores with a more efficient and lightweight model structure. In summary, our proposed HTR system not only showcases the potential of integrating self-attention blocks into a standard CNN-RNN architecture but also emphasizes the importance of model efficiency in practical applications. Future work will focus on optimizing the architecture further and addressing the limitations identified in this work, aiming to push the possibility boundaries in efficient HTR systems.

Table 7. Ablation on the components of the model Encoder and Head.

		CER(%)↓		WER(%)↓	
Encoder	**Head**	**Val**	**Test**	**Val**	**Test**
CNN	BiLSTM	3.31 ± 0.07	4.65 ± 0.03	12.19 ± 0.35	15.85 ± 0.05
CNN	Transformer	4.10 ± 0.06	5.89 ± 0.04	15.28 ± 0.36	14.64 ± 0.15
CNN+MobileViT	BiLSTM	3.09 ± 0.03	4.29 ± 0.08	11.43 ± 0.09	14.64 ± 0.15

Table 8. Line-level recognition performance comparison for IAM and RIMES datasets.

			IAM		RIMES	
Method	**# params**	**synth**	**CER(%)↓**	**WER(%)↓**	**CER(%)↓**	**WER(%)↓**
Dutta et al. [7]	-	✗	5.80	17.80	5.07	14.70
Michael et al. [23]	-	✗	5.24	-	-	-
Tassopoulou et al. [29]	-	✗	5.18	17.68	-	-
Yousef et al. [35]	-	✗	4.90	-	-	-
Retsinas et al. [26]	-	✗	4.55	16.08	3.04	10.56
Retsinas et al. [25]	5.7M	✗	4.62	15.80	2.75	9.93
Coquenet et al. [5]	2.7M	✗	4.54	14.55	2.15	6.72
Attention-based Architectures						
Kang et al. [16]	100M	✗	7.62	24.54	-	-
Barrere et al. [1]	-	✗	7.42	29.09	-	-
Wick et al. [32]	13M	✗	6.02	-	-	-
Barrere et al. [2]	6.9M	✗	5.70	18.86	-	-
Wang et al. [31]		✓	6.40	19.60	2.70	8.90
Wick et al. [32]	27M	✗	5.67	-	-	-
Barrere et al. [1]	-	✗	5.07	21.47	-	-
Barrere et al. [2]	6.9M	✓	4.76	16.31	-	-
Kang et al. [16]	100M	✓	4.67	15.45	-	-
$\text{TrOCR}_{\text{SMALL}}$ [18]	62M	✓	4.22	-	-	-
$\text{TrOCR}_{\text{BASE}}$ [18]	334M	✓	3.42	-	-	-
$\text{TrOCR}_{\text{LARGE}}$ [18]	558M	✓	2.89	-	-	-
Diaz et al. [6]	105M	✓	2.75	-	1.99	-
DTrOCR [10]	105M	✓	2.38	-	-	-
Proposed	10.9M	✗	4.22	14.58	2.70	9.46

5 Discussion

This work acts as an intermediate step to well-designed lightweight HTR architectures that harness the prowess of modern Transformer Architectures. Even though the attention modules used in this paper are limited, we have seen non-trivial improvement aiming to provide global information from the typical self-attention rationale. Further experimentation could shed light to the intuition behind such improvement. For example, the global information, exchanged in the attention modules, could be aligned with style, hinting towards writer dependent features. We also raise the question if you should blindly go towards very large Transformer architectures. The next logical step, as future work, is to decompose the vision module from the language modeling aspect, where transformers excel - as seen by [18] and [10], and explore if such a research path leads to more compact architectures.

6 Conclusion

In this paper, we successfully demonstrated the integration of Transformer modules into established CRNN architectures for HTR. We proposed a hybrid CNN-MobileViT image encoder that effectively balances efficiency and performance. In our analysis, we argued that while Transformers are a powerful learning module, proper integration within an HTR pipeline is not straightforward and requires careful adaptation to obtain its full capabilities. By deploying Transformer modules as part of the image understanding module, we manage to achieve state-of-the-art performance, all the while avoiding presupposing a constrained "resource-hungry" regime. Through extensive experiments and ablation studies, we highlighted the efficacy of our model, which in many cases even outperforms existing Transformer-based systems that are higher in parameters and dependent on synthetic data.

References

1. Barrere, K., Soullard, Y., Lemaitre, A., Coüasnon, B.: Transformers for Historical Handwritten Text Recognition. In: Doctoral Consortium-ICDAR 2021 (2021)
2. Barrere, K., Soullard, Y., Lemaitre, A., Coüasnon, B.: A Light Transformer-Based Architecture for Handwritten Text Recognition. In: International Workshop on Document Analysis Systems. pp. 275–290. Springer (2022)
3. Bhunia, A.K., Khan, S., Cholakkal, H., Anwer, R.M., Khan, F.S., Shah, M.: Handwriting Transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1086–1094 (2021)
4. 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 Transactions on Pattern Analysis and Machine Intelligence **33**(10), 2066–2080 (2011)

5. Coquenet, D., Chatelain, C., Paquet, T.: End-to-end Handwritten Paragraph Text Recognition Using a Vertical Attention Network. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(1), 508–524 (2022)
6. Diaz, D.H., Qin, S., Ingle, R.R., Fujii, Y., Bissacco, A.: Rethinking Text Line Recognition Models. ArXiv **abs/2104.07787** (2021)
7. Dutta, K., Krishnan, P., Mathew, M., Jawahar, C.V., Dutta, K., Krishnan, P., Mathew, M.: Improving CNN-RNN Hybrid Networks for Handwriting Recognition. 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR) pp. 80–85 (2018)
8. d'Arce, R., Norton, T., Hannuna, S., Cristianini, N.: Self-attention Networks for Non-recurrent Handwritten Text Recognition. In: International Conference on Frontiers in Handwriting Recognition. pp. 389–403. Springer (2022)
9. Espana-Boquera, S., Castro-Bleda, M.J., Gorbe-Moya, J., Zamora-Martinez, F.: Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(4), 767–779 (2010)
10. Fujitake, M.: DTrOCR: Decoder-only Transformer for Optical Character Recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 8025–8035 (2024)
11. Giménez, A., Khoury, I., Andrés-Ferrer, J., Juan, A.: Handwriting word recognition using windowed Bernoulli HMMs. Pattern Recognition Letters **35**, 149–156 (2014)
12. 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)
13. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems **28**(10), 2222–2232 (2016)
14. 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)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
16. Kang, L., Riba, P., Rusiñol, M., Fornés, A., Villegas, M.: Pay attention to what you read: Non-recurrent handwritten text-line recognition. Pattern Recognition **129**, 108766 (2022)
17. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On large-batch training for deep learning: Generalization gap and sharp minima (2017)
18. Li, M., Lv, T., Cui, L., Lu, Y., Florêncio, D.A.F., Zhang, C., Li, Z., Wei, F.: TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. In: AAAI Conference on Artificial Intelligence (2021)
19. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. In: International Conference on Learning Representations (2017)
20. Markou, K., Tsochatzidis, L.: A convolutional recurrent neural network for the handwritten text recognition of historical greek manuscripts. In: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10-15, 2021, Proceedings, Part VII. pp. 249–262. Springer (2021)
21. Marti, U.V., Bunke, H.: The IAM-database: an English sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition **5**, 39–46 (2002)

22. Mehta, S., Rastegari, M.: MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer. In: International Conference on Learning Representations (2021)
23. 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)
24. 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)
25. Retsinas, G., Sfikas, G., Gatos, B., Nikou, C.: Best Practices for a Handwritten Text Recognition System. In: International Workshop on Document Analysis Systems. pp. 247–259. Springer (2022)
26. Retsinas, G., Sfikas, G., Nikou, C., Maragos, P.: Deformation-Invariant Networks For Handwritten Text Recognition. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 949–953 (2021). https://doi.org/10.1109/ICIP42928.2021.9506414
27. Retsinas, G., Sfikas, G., Nikou, C., Maragos, P.: From Seq2Seq Recognition to Handwritten Word Embeddings. In: Proceedings of the British Machine Vision Conference (BMVC) (2021)
28. Sfikas, G., Retsinas, G., Dimitrakopoulos, P., Gatos, B., Nikou, C.: Shared-operation hypercomplex networks for handwritten text recognition. In: International Conference on Document Analysis and Recognition. pp. 200–216. Springer (2023)
29. Tassopoulou, V., Retsinas, G., Maragos, P.: Enhancing handwritten text recognition with n-gram sequence decomposition and multitask learning. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 10555–10560. IEEE (2021)
30. Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is All you Need. In: Neural Information Processing Systems (2017)
31. Wang, T., Zhu, Y., Jin, L., Luo, C., Chen, X., Wu, Y., Wang, Q., Cai, M.: Decoupled Attention Network for Text Recognition. In: AAAI Conference on Artificial Intelligence (2019)
32. Wick, C., Zöllner, J., Grüning, T.: Transformer for Handwritten Text Recognition Using Bidirectional Post-decoding. In: International Conference on Document Analysis and Recognition. pp. 112–126. Springer (2021)
33. Wick, C., Zöllner, J., Grüning, T.: Rescoring sequence-to-sequence models for text line recognition with CTC-prefixes. In: International Workshop on Document Analysis Systems. pp. 260–274. Springer (2022)
34. Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollár, P., Girshick, R.: Early Convolutions Help Transformers See Better. Advances in Neural Information Processing Systems **34**, 30392–30400 (2021)
35. Yousef, M., Hussain, K.F., Mohammed, U.S.: Accurate, data-efficient, unconstrained text recognition with convolutional neural networks. Pattern Recognition **108**, 107482 (2020)
36. Zhang, X., Su, Y., Tripathi, S., Tu, Z.: Text Spotting Transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9519–9528 (2022)