Dual Downsample Vision Transformer for Handwritten Text Recognition

Yew Lee Tan[1], Ernest Yu Kai Chew[1], Jung-jae Kim[2], and Adams Wai-Kin Kong[1]

[1] Nanyang Technological University, Singapore
[2] Institute for Infocomm Research, A*STAR, Singapore

Abstract. Line-level handwritten text recognition (HTR) transcribes text from scanned or photographed documents, with transformer-based models recently achieving strong performance. However, these models typically require extensive real and synthetic datasets and significant computational resources, rendering them impractical for tasks involving small, private datasets. The lack of open-source additional data further hinders fair performance comparisons. This work proposes a Dual Downsample Vision Transformer (DDViT) for HTR, relying solely on small, open-source datasets like IAM. Building on an efficient vision transformer baseline, DDViT generates two predictions via distinct downsampling layers, selecting the final output with a novel dual maps scoring method. DDViT achieves state-of-the-art character error rates (e.g., 3.34% on IAM-A, a 1.5% point improvement over prior works) across nine related studies and remains competitive with methods using additional data, demonstrating effective HTR without resource-intensive requirements.

Keywords: Handwritten text recognition · Transformer · Downsample.

1 Introduction

Line-level HTR aims to digitise textual information from line-level images of handwritten document. It is a sub-field of HTR with applications in digital storage and retrieval of handwritten forms, doctors' memos, etc. It can be viewed as the most studied sub-domain of HTR as it can be used standalone as well as being integrated into page-level HTR [5].

Generally, an encoder-decoder pipeline is employed in HTR models where the encoder encodes a line image into features and the decoder decodes image features into a string of characters. Early architectures were often made up of convolutional neural network (CNN) encoder followed by recurrent neural network (RNN) decoder [11, 26].

Over the years, many works have also begun to incorporate attention mechanism into their HTR models [3, 4, 12, 16, 17, 25]. Poulos and Valle [24] studied and utilised various attention mechanisms in RNN decoder for line-level HTR. Similarly, Michael et al. [23] proposed a CNN-RNN encoder with an attention

based RNN decoder. However, these models are overshadowed by transformer-based models in recent years as transformer [31], with its attention mechanism, attained competitive results in various domains such as ViT [9] in computer vision tasks.

Despite their success, transformer-based HTR models often rely on extensive real and synthetic datasets and substantial computational resources, posing challenges for applications limited to small, private datasets. For instance, the top-performing model by Li et al. [19] uses 558 million parameters and 700 million additional samples, trained on 32 GPUs – resources inaccessible to many. Moreover, the proprietary nature of these additional datasets hinders standardized performance evaluation. This work addresses these limitations by developing a transformer-based HTR approach that operates solely on small, open-source datasets like IAM. We demonstrate that transformers can achieve competitive HTR performance without large datasets, challenging the prevailing reliance on extensive data.

Many text recognition studies, including HTR, use multiple predictions to improve accuracy, often relying on resource-intensive methods like ensemble models or dual decoders [1, 27, 29, 33, 34]. For example, Wick et al. [31] employed two decoders and a voting scheme, significantly increasing computational costs. In contrast, we propose a more efficient solution with our Dual Downsample Vision Transformer (DDViT), which builds on a transformer baseline. DDViT generates two predictions using distinct downsampling layers, and a novel dual maps scoring method selects the final output, boosting accuracy with minimal increases in parameters and inference time compared to the baseline.

In summary, the contribution is as follows: (1) an encoder-only baseline model made up of vision transformer, (2) dual downsample vision transformer which improves the baseline model, (3) a dual maps scoring method which scores the predictions, and (4) state-of-the-art results of at least 1% point improvement on character error rate (CER) against 9 related works trained with different splits of IAM dataset.

The rest of the paper is organised as follows. Sec. 2 explores the related works within the scope of this study. Sec. 3 presents proposed models and scoring methods. Sec. 4 discusses the results and ablation studies while Sec. 5 concludes this work.

2 Related Works

Handwritten text recognition (HTR) models have traditionally relied on convolutional neural networks (CNNs) and recurrent neural networks (RNNs), particularly long short-term memory (LSTM) units. For instance, Yousef et al. [36] utilized a CNN-only architecture for HTR, while Michael et al. [23] proposed a hybrid approach combining a CNN-LSTM encoder with an attention-based LSTM decoder. Similarly, Coquenet et al. [6] introduced the Vertical Attention Network, featuring a CNN encoder and LSTM decoder, capable of performing both line-level and paragraph-level recognition. Kizilirmak et al. [18]

also adopted a CNN-LSTM framework, pre-training their model on a synthetic dataset of 2.5 million text line images. However, Wang et al. [32] identified a limitation in attention-based mechanisms, noting frequent misalignment issues, and proposed a convolution alignment module to mitigate this problem.

The advent of transformers has shifted the focus in HTR research toward more advanced architectures. Kang et al. [15] developed a CNN-transformer hybrid encoder paired with a transformer decoder, complemented with 138,000 synthetic samples. Wick et al. [34] similarly enhanced their transformer-based model with a private dataset of 1 million real and synthetic samples. Diaz et al. [8] employed a variant of the fused inverted bottleneck [28] integrated with a transformer, supported by a custom dataset comprising synthetic text and handwritten document images. Li et al. [19] took a different approach, using a vanilla transformer encoder-decoder architecture pre-trained in two stages: first on 684 million text line images extracted from 2 million internet-sourced PDF files, followed by a second stage on 17 million synthetic text images. Their best-performing model, with 558 million parameters, was trained on 32 GPUs with 32GB memory each.

Transformer-based HTR models often leverage large, self-collected datasets of real and synthetic data to boost performance. However, these datasets are typically proprietary, making standardized evaluation across studies difficult. Moreover, the computational resources required—such as extensive GPU setups—may be prohibitive for many researchers. Additionally, the potential of transformers in line-level HTR without supplementary data remains underexplored. To address these gaps, this work proposes a robust transformer baseline trained solely on a small, open-source dataset.

Another common strategy in text recognition involves using multiple predictions to enhance accuracy, albeit at the expense of increased computational overhead. For example, Wemhoener et al. [33] trained multiple models on different editions of a book, combining their outputs via a multiple sequence alignment framework. Similarly, Reul et al. [27] trained multiple models on distinct dataset partitions, employing a voting scheme to determine the final prediction. In the HTR domain, Wick et al. [34] utilized two decoders operating in opposite directions, with voting and heuristics guiding the final output. While effective, these approaches significantly increase inference time and model complexity.

3 Approach

3.1 Model Overview

Illustrations of a baseline model and proposed dual downsample vision transformer (DDViT) are shown in Fig. 1. The baseline model, adapted from ViT [9], is made up of L transformer layers followed by a downsampling layer. An input image is first resized into $H_{resized} \times W_{resized}$ and then split into patches of dimension $D_{patch} = H_{patch} \times W_{patch}$ resulting in N_{patch} patches. A linear projection is used to project D_{patch} to D_{emb} and a positional encoding is then added.

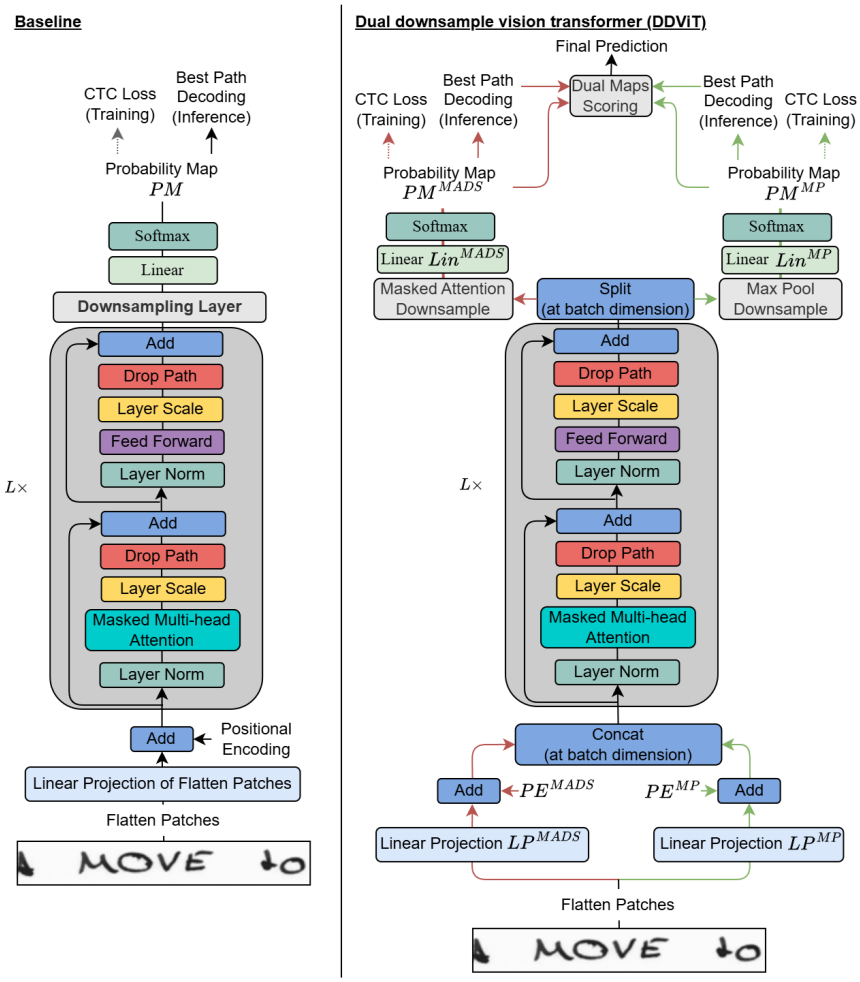


Fig. 1. Architectures of the baseline model and the proposed Dual Downsample Vision Transformer (DDViT). The two parallel downsampling components, Max Pooling (MP) and Masked Attention Downsampling (MADS), are applied to each output which are scored using a dual maps scoring method.

Following which, the sequence of embeddings will be passed into transformer encoder layers with masked multi-head attention. Using a mask for multi-head attention (MHA) in transformer encoder to restrict attention region was first introduced in NLP tasks [37]. Different from a vanilla ViT, the embeddings will go through layer scale and drop path adapted from the work by Touvron et al. [30]. The layer scale contains D_{emb} learnable weights which scale the embeddings element-wise. The drop path is an implementation of stochastic depth [14] by Touvron et al. [30] for ViT.

After the transformer layers, the embeddings sequence is then passed to a downsampling layer which downsamples the sequence length from N_{patch} to downsampled length of $N_{ds} = \frac{W_{resized}}{W_{patch}}$. Max pooling (MP) and a proposed masked attention downsampling (MADS) can be used as downsampling layer. The downsampled output will go through a linear layer which projects D_{emb} to D_{class} where D_{class} is the number of output classes. Softmax is then applied to produce a probability map. CTC loss [10] was used to train the models which produce character predictions non-autoregressively.

Different from baseline, DDViT has 2 downsampling layers resulting in 2 outputs for a given image as seen in Fig. 1. A separate linear projection layer, positional encoding, and output linear layer cater to the respective MADS and MP downsampling. The downsampling layers produce 2 probability maps followed by CTC best path decoding predictions. They are used to calculate the prediction scores through a dual map scoring method for choosing the final prediction. The concatenation at the batch dimension prior to the transformer layers is designed to lower the inference time since both downsampling methods share the same weights in transformer layers.

For the experiments, the images were resized to $H_{resized} \times W_{resized}$ of 32×768, and the patch resolution $H_{patch} \times W_{patch}$ was 8×4. L was set at 18, D_{emb} was 256 with 8 heads for multi-head attention while the intermediate dimension in feed forward layer was 1024. N_{ds} was 192 and D_{class} was dependent on the dataset used. The drop rate of drop path was set at 0.1 and the initialisation value for layer scale was 0.01. The models were trained for 2000 epochs on RTX 3090 with a batch size of 64 and learning rate of 0.001 on AdamW optimiser. Cosine scheduler was used with 5 warm up epochs and a warm up and minimum learning rates of $1e-6$ and $1e-5$ respectively. Augmentation methods used for training followed the work by Coquenet et al. [6].

3.2 Masking

Masking in text recognition has been explored in recent works such as MaskOCR [21] and HTR-VT [20], where random masking is applied primarily during training, following the BERT-style masked language modeling [7]. In contrast, our masking strategy is inspired by the structured attention constraints introduced in NLP models such as BigBird [37]. It is deterministic and spatially structured, and is applied during both training and inference to explicitly restrict the attention span. This design helps mitigate overfitting by limiting the receptive field, thereby improving generalization, as evidenced in Sec. 4.3. An illustration of mask construction for MHA is shown in Fig. 2. An image is first 'patchified' into $N_{row} = \frac{H_{resized}}{H_{patch}}$ rows and $N_{col} = \frac{W_{resized}}{W_{patch}}$ columns of image patches. The $N_{patch} = N_{row}N_{col}$ image patches are then flatten in column-major order. A mask is constructed for each patch, $n_{patch} \in [0, 1, 2, ..., N_{patch} - 1]$.

In the top of figure, the number in red denotes n_{patch} while the number in blue denotes the column number n_{col}. The patches are flatten in column major order. Patches belonging to the same column n_{col} (labeled in blue) share the same

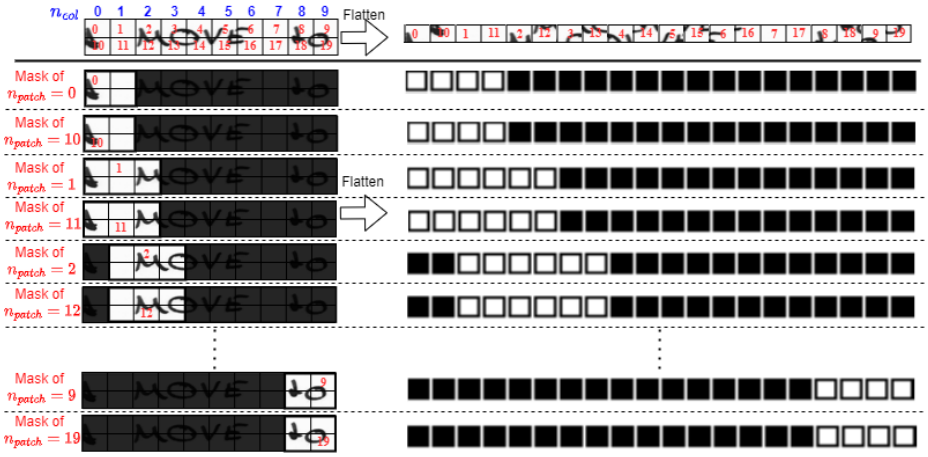


Fig. 2. Illustration of mask construction for multi-head attention. The number in red denotes n_{patch}, while the number in blue denotes n_{col}.

mask configuration where $n_{col} \in [0, 1, 2, ..., \frac{W_{resized}}{W_{patch}} - 1]$ and $W_{resized}$ and W_{patch} are the width of image and patch respectively. For instance, $n_{patch} = \{0, 10\}$ use the same mask as illustrated in the top portion of Fig. 2. After the flattening, the masks can be viewed as having window of size W_{mask} with stride of N_{row}. In the illustration, $W_{mask} = 6$ with stride 2.

The attention mask for a given n_{patch} at column n_{col} and row n_{row} will attend to patches from columns $\max(0, n_{col} - 16)$ to $\min(n_{col} + 16, N_{col} - 1)$ and from $n_{row} = 0$ to $n_{row} = N_{row} - 1$ of the 'patchified' image. Patches that belongs to the same column use the same configuration of mask. Therefore, there are a total of N_{col} number of different attention masks. For the experiments, $N_{col} = 192$, $N_{row} = 4$, $N_{patch} = 768$, $W_{mask} = 132$, stride $= 4$. Hence, patch numbers $n_{patch} = \{0, 1, 2, 3\}$ belong to the first column of the 'patchified' image, $n_{patch} = \{4, 5, 6, 7\}$ belong to the second column and so on.

3.3 Downsampling Layer

Max pooling (MP), and a proposed masked attention downsampling (MADS), were used in DDViT. For MP, the window size was 132 with a stride of 4 and padding 64.

The MADS layer is illustrated in Fig. 3. The output $\mathbf{x_{out}} \in \mathbb{R}^{N_{patch} \times D_{emb}}$ of transformer layers will go through 2 separate linear layers to produce $\mathbf{A} \in \mathbb{R}^{N_{patch} \times N_{ds}}$ and $\mathbf{V} \in \mathbb{R}^{N_{patch} \times D_{emb}}$ as per Eq. (1) and Eq. (2). Masking and softmax are then applied on $\mathbf{A}$ to form an attention map. The attention map will then act on $\mathbf{V}$ which results in a downsampled output of $\mathbb{R}^{N_{ds} \times D_{emb}}$ as in

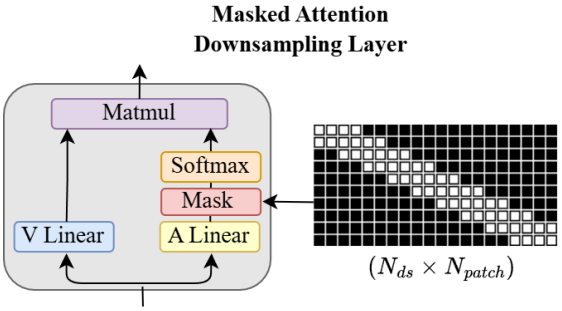


Fig. 3. MADS, where the illustrated mask is of window size 6 and stride 2.

Eq. (3). The mask with window size 132 and stride of 4 was used for MADS.

$$\mathbf{A} = \mathbf{x_{out}}\mathbf{B_A} + \mathbf{c_A}, \tag{1}$$

where $\mathbf{B_A} \in \mathbb{R}^{D_{emb} \times N_{ds}}$, $\mathbf{c_A} \in \mathbb{R}^{N_{ds}}$.

$$\mathbf{V} = \mathbf{x_{out}}\mathbf{B_V} + \mathbf{c_V}, \tag{2}$$

where $\mathbf{B_V} \in \mathbb{R}^{D_{emb} \times D_{emb}}$, $\mathbf{c_V} \in \mathbb{R}^{D_{emb}}$.

$$\text{downsampled output} = \text{softmax}(\text{mask}(\mathbf{A}^\top))\mathbf{V}. \tag{3}$$

3.4 Scoring Methods

Dual Maps Scoring The proposed models were trained using CTC loss [10] which requires an additional class for 'blank' denoted by '⟨-⟩'. Given a sequence of image patches $\mathbf{x}$ of length N_{patch}, the baseline outputs a probability map $\mathbf{PM} \in \mathbb{R}^{N_{ds} \times D_{class}}$. Let $\mathbf{y} = (y_0, y_1, y_2, ..., y_{N_{ds}-1})$ be an arbitrary sequence of characters and PM_{i,y_i} be the probability of observing y_i at i where $i \in \{0, 1, 2, ..., N_{ds}\}$. Therefore, the probability of observing $\mathbf{y}$ is as follows:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=0}^{N_{ds}-1} PM_{i,y_i}. \tag{4}$$

Given the ground-truth character sequence of a sample, $\mathbf{z}$, with length $N_z \leq N_{ds}$, there are multiple output sequences $\mathbf{y}$ that can align with the ground-truth. Let $\mathbf{B}$ be a many-to-one mapping function that removes blanks and repeated characters in $\mathbf{y}$. Then, the possible alignments if $N_{ds} = 3$ and $\mathbf{z}$ ='ab' are: $\mathbf{B}$('aab') = $\mathbf{B}$('abb') = $\mathbf{B}$('⟨-⟩ab') = $\mathbf{B}$('a⟨-⟩b') = $\mathbf{B}$('ab⟨-⟩') ='ab'.

CTC loss then maximises the probability of the sum of all possible alignments of $\mathbf{z}$ as follows:

$$p(\mathbf{z}|\mathbf{x}) = \sum_{\mathbf{y} \in \mathbf{B}^{-1}(\mathbf{z})} p(\mathbf{y}|\mathbf{x}). \tag{5}$$

As each downsampling layer of DDViT produces a probability map, there will be 2 maps denoted by $\mathbf{PM}^{MP}$ and $\mathbf{PM}^{MADS}$ from MP and the proposed

MADS which result in 2 output character sequences $\mathbf{y}^{MP}$ and $\mathbf{y}^{MADS}$. Using best path decoding, the character with the highest probability at each i is chosen from the respective probability map following Eq. (6) and Eq. (7) where $PM^{MP}_{i,:}$ and $PM^{MADS}_{i,:} \in \mathbb{R}^{D_{class}}$ are the i row vector of the respective $\mathbf{PM}^j, j \in \{MP, MADS\}$.

$$y_i^{MP} = \text{argmax}(PM^{MP}_{i,:}), \tag{6}$$

$$y_i^{MADS} = \text{argmax}(PM^{MADS}_{i,:}). \tag{7}$$

Following which, the predicted strings are $\mathbf{s}^{MP} = \mathbf{B}(\mathbf{y}^{MP})$ and $\mathbf{s}^{MADS} = \mathbf{B}(\mathbf{y}^{MADS})$. The probability of observing the strings follows Eq. (8) where $j = k = MP$ for the probability of $\mathbf{s}^{MP}$ calculated from $\mathbf{PM}^{MP}$, and $j = k = MADS$ for $\mathbf{s}^{MADS}$ calculated from $\mathbf{PM}^{MADS}$.

$$p^j(\mathbf{s}^k|\mathbf{x}) = \sum_{\mathbf{y} \in \mathbf{B}^{-1}(\mathbf{s}^k)} \prod_{i=0}^{N_{ds}-1} PM^j_{i,y_i}, \tag{8}$$

$$\text{where } k \in \{MP, MADS\}.$$

The proposed dual map scoring method scores $\mathbf{s}^{MP}$ and $\mathbf{s}^{MADS}$ as shown in Eq. (9) and Eq. (10) where $\mathbf{s}^k$ with the highest score is chosen as the final prediction.

$$\text{MP Score} = p^{MP}(\mathbf{s}^{MP}|\mathbf{x}) + p^{MADS}(\mathbf{s}^{MP}|\mathbf{x}), \tag{9}$$

$$\text{MADS Score} = p^{MP}(\mathbf{s}^{MADS}|\mathbf{x}) + p^{MADS}(\mathbf{s}^{MADS}|\mathbf{x}). \tag{10}$$

Best Path Scoring Best path scoring method scores the strings $\mathbf{s}^{MP}$ and $\mathbf{s}^{MADS}$ based on the best path as shown in Eq. (11) and Eq. (12). The string with the highest score will be chosen as the final prediction.

$$\text{MP Score} = \prod_{i=0}^{N_{ds}-1} \max(PM^{MP}_{i,:}), \tag{11}$$

$$\text{MADS Score} = \prod_{i=0}^{N_{ds}-1} \max(PM^{MADS}_{i,:}). \tag{12}$$

String Scoring String scoring method scores $\mathbf{s}^{MP}$ and $\mathbf{s}^{MADS}$ based on the probability of predicting the strings following Eq. (8). The calculation of scores are shown in Eq. (13) and Eq. (14).

$$\text{MP Score} = p^{MP}(\mathbf{s}^{MP}|\mathbf{x}), \tag{13}$$

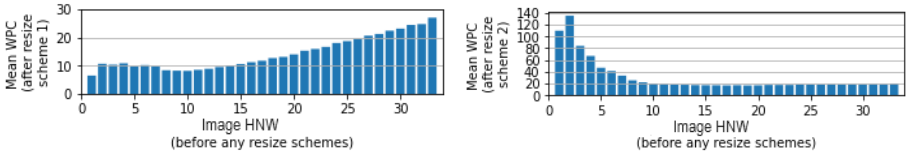
$$\text{MADS Score} = p^{MADS}(\mathbf{s}^{MADS}|\mathbf{x}). \tag{14}$$

3.5 Resize and Repeat

As handwriting differs between individuals, the average image width a character occupies varies. The distribution of width per character (WPC) of images in the open-source HTR dataset may be a contributing factor in recognition accuracy. Therefore, a preprocessing method that repeats the images was introduced to create a more uniform WPC distribution and improve accuracy.

Generally, there are two schemes to resize an image. Scheme 1 involves resizing the image to a fixed height while preserving its aspect ratio, followed by padding to reach a target width. Scheme 2, in contrast, resizes the image directly to a fixed height and width, potentially distorting the original aspect ratio.

Fig. 4 shows the mean width per character (WPC) against height normalised width (HNW) of images from train dataset. The HNW is the width of original image where height is set to 1. It is computed before any resizing schemes are applied as illustrated in Fig. 5. This serves as a categorization ensuring that images belong to the same HNW bin regardless of the type of resize scheme used. Doing so will allow visualisation on the impact of resizing scheme on WPC. The WPC of the image is calculated after resize by dividing resized width with number of ground-truth characters. Therefore, different resizing schemes will result in different WPC for a given image.



(a) (b)

Fig. 4. Mean WPC of training images with respect to their HNW. (a) Images were resized using scheme 1 with height 32 and varying width to maintain aspect ratio. (b) Images were resized with scheme 2 with fixed height and width of 32 and 768.

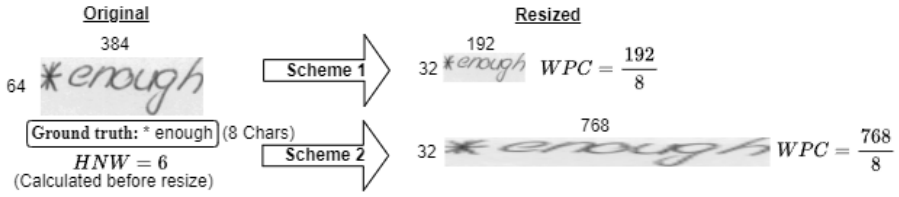


Fig. 5. An example on calculation of HNW and WPC.

It can be seen in Fig. 4 (a) that for images with HNW lesser than 10, the mean WPC fluctuates at about 10. Above HNW of 10, the mean WPC increases from about 10 to 30. Therefore, it is likely that a longer HNW of image signifies wider handwriting. Overall, if scheme 1 is used, the WPC increases with HNW and this characteristic may cause the model to have weaker generalisation due to the small training dataset. On the other hand, as shown in Fig. 4 (b) for scheme 2, the mean WPC is more uniform at about 20 for images with HNW of above 10. However, below that, the mean WPC varied greatly from 20 to 140.

To deal with this issue and enhance scheme 2, the repeating image method is proposed where images are repeated as shown in Fig. 6 which lowers the WPC. Repeated images are separated by black pixels while repeated ground-truth texts are separated by a special token ⟨end⟩. Fig. 7 shows the mean WPC of scheme 2 with repeat image method. The mean is considerably more uniform at about 20 to 30 as compared to Fig. 4 (b). The repeating of image is based upon the statistics of training dataset.

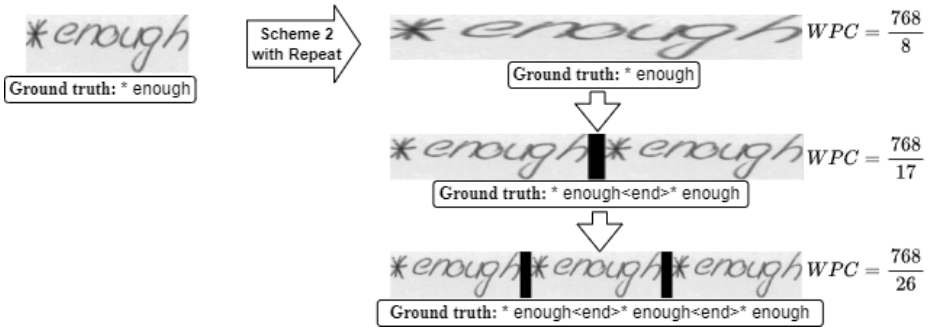


Fig. 6. Illustration of WPC decreasing with repeating image.

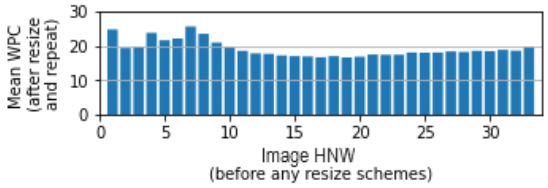


Fig. 7. Mean WPC with respect to HNW. Scheme 2 and repeating image were used.

For the repeat method, the statistics of WPC are first calculated with scheme 2. Images with $HNW = i$, where $i \leq 12$ are used to calculate the mean of WPC, μ_i, and its standard deviation, σ_i. Whereas the rest of the images in the dataset

(with HNW greater than 12) are used to compute μ_T and σ_T of a target WPC_T. During training, for a given image sample with $HNW = i \leq 12$, a WPC_i is drawn from $\mathcal{N}(\mu_i, \sigma_i^2)$ and a WPC_T is drawn from $\mathcal{N}(\mu_T, \sigma_T^2)$. The image is then repeated until WPC_i decreases to WPC_T. An example of WPC decreasing with repeating image is illustrated in Fig. 6. The reason why WPC_i and WPC_T are drawn from distributions is to introduce randomness during training. For inference, $WPC_i = \mu_i$ and $WPC_T = \mu_T$. Only the predictions before $\langle end \rangle$ are kept. The analysis of effectiveness of this preprocessing method is provided in section 1 of supplementary material.

4 Experiments and Results

4.1 Dataset

The baseline models and DDViT were trained and evaluated on line-level IAM dataset [22] and RIMES dataset [2]. IAM dataset is an English handwritten dataset which may be the only open-source, non-historical line-level handwritten dataset for English [18]. Michael et al. [23] highlighted that different splits of IAM dataset were used in various works causing unfair comparisons. The authors identified 3 different versions of splits as follows. IAM-A consisting of 6161 training samples, 966 validation samples, and 2915 testing samples. IAM-B containing 6482 training, 976 validation, and 2915 testing samples. Lastly, IAM-C with 6161 training, 940 validation, and 1861 testing samples. Rimes dataset is a modern french dataset consisting of 10188 training, 1138 validation, and 778 testing samples. Validation set was used in various experiments for hyperparameter tuning. It was then used together with training set to train the best performing models.

4.2 Comparison with State-of-the-Art Methods

Related Works without Additional Data Tab. 1 compares the results of proposed models with related works where no external n-grams language model and no additional synthetic or self-collected real data were used. The proposed models were trained on each split/dataset separately where '(+V)' denotes model trained with validation set. Both 'DDViT' and 'DDViT (+V)' outperformed all related works in IAM splits. The best results for all IAM splits were attained by 'DDViT (+V)' with a CER of 3.34% for IAM-A (lower than next best related work by 1.53% points), CER of 3.37% for IAM-B (lower than next best related work by 1.17% points), and CER of 2.56% for IAM-C (lower than next best related work by 4.10% points). For RIMES dataset, 'DDViT' attained the second best results for CER as compared to 5 other related works where the best performing work by Coquenet et al. [6] made use of paragraph-level image data on top of line-level images. These suggest that a transformer-only model trained on a small dataset is able to produce competitive results. Fig. 8 illustrates an example of a success case of DDViT with dual maps scoring while Fig. 9 shows an example of a failure case.

Table 1. Comparison of CER and WER on IAM splits and RIMES with various works. (+V) signifies the use of both train and validation set in final training and (+P) is the use of paragraph-level data in training. The best results from others works are underlined. Values in the parenthesis are the difference in percentage points between the results of proposed models as compared to the best results from related works.

Method	Yr.	IAM-A		IAM-B		IAM-C		RIMES	
		CER (%)	WER (%)	CER (%)	WER (%)	CER (%)	WER (%)	CER (%)	WER (%)
Michael et al. [23]	ICDAR'19	5.24	-	-	-	-	-	-	-
Michael et al. (+V) [23]	ICDAR'19	4.87	-	-	-	-	-	-	-
Wang et al. [32]	AAAI'20	-	-	6.40	19.6	-	-	2.70	8.90
Yousef et al. [36]	PR'20	4.90	-	-	-	-	-	-	-
Wick et al. [34]	ICDAR'21	5.38	-	-	-	-	-	-	-
Wick et al. [35]	'IWDAS'22	5.09	15.88	-	-	-	-	3.88	-
Kang et al. [15]	PR'22	-	-	7.62	24.54	-	-	-	-
Coquenet et al. [6]	TPAMI'22	-	-	4.97	16.31	-	-	3.08	8.14
Coquenet et al. (+P) [6]	TPAMI'22	-	-	4.54	14.55	-	-	2.15	6.72
Kizilirmak et al. [18]	'23	5.20	14.86	-	-	-	-	-	-
Hamdam et al. [13]	IJDAR'23	-	-	-	-	6.66	18.97	4.24	10.03
DDViT		3.79 (-1.08)	12.66 (-2.20)	3.73 (-0.81)	12.38 (-2.17)	3.82 (-2.84)	12.11 (-6.86)	2.67 (+0.52)	9.44 (+2.72)
DDViT (+V)		3.34 (-1.53)	11.11 (-3.75)	3.37 (-1.17)	11.09 (-3.46)	2.56 (-4.10)	7.78 (-11.19)	2.53 (+0.38)	8.60 (+1.88)

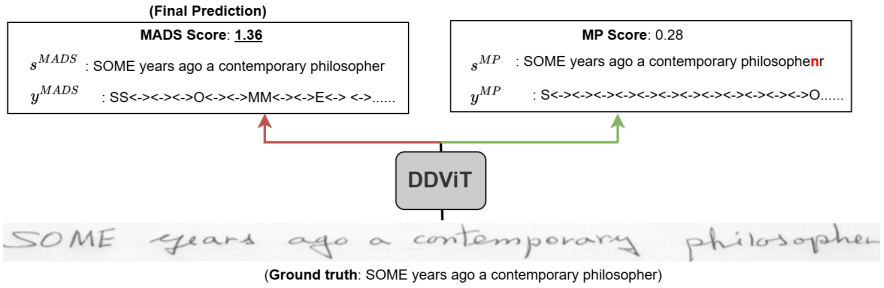


Fig. 8. An example of a success case.

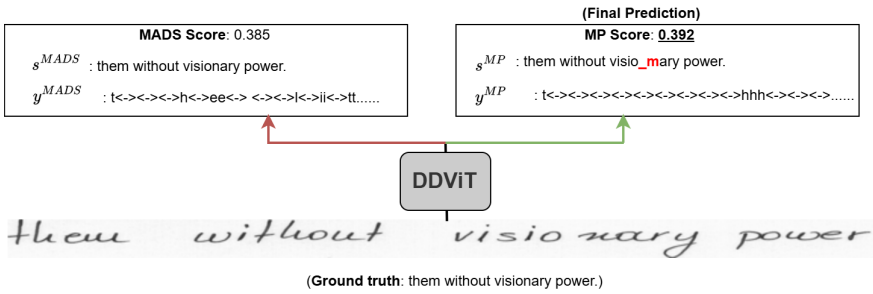


Fig. 9. An example of a failure case.

Related Works with Additional Data To clarify the distinction between evaluation settings, Tab. 1 presents results from methods trained *without* any

additional synthetic or real data, while Tab. 2 includes methods that *do* utilize such data. This separation is designed to ensure a fair comparison among methods operating under similar data constraints.

Many high-performing methods, such as TrOCR [19], rely heavily on large-scale pretraining and extensive synthetic datasets. $\text{TrOCR}_{\text{LARGE}}$, for instance, is trained on over 700 million text line images and leverages a 558M-parameter model across 32 GPUs. While such resources lead to strong performance, they are often inaccessible to many researchers or practitioners. Similarly, Diaz et al. [8] and Wick et al. [35] used an undisclosed amount of synthetic and self-collected real images, while Kang et al. [15] and Kizilirmak et al. [18] employed 138 thousand and 2.5 million synthetic images, respectively. The results of these methods (trained with additional data) are reported in Tab. 2.

In contrast, DDViT (+V) achieves competitive performance without using any additional synthetic or real data, relying solely on small, open-source datasets such as IAM. As shown in Tab. 2, DDViT (+V) outperforms three other approaches and remains reasonably close to $\text{TrOCR}_{\text{Large}}$ in accuracy, despite the absence of large-scale pretraining or massive datasets. While it does not surpass $\text{TrOCR}_{\text{Large}}$, this comparison highlights the efficiency and practicality of DDViT in low-resource settings.

Table 2. Comparison of CER and WER on IAM-A and IAM-B with methods that utilize additional synthetic and/or real data beyond IAM. Despite not using such resources, DDViT (+V) remains competitive.

Method	**Year**	**Add. Data (mil.)**	**IAM-A**		**IAM-B**	
			CER (%)	WER (%)	CER (%)	WER (%)
Diaz et al. [8]	'21	> 0	2.96	-	-	-
Kang et al. [15]	PR'22	0.14	-	-	4.67	15.45
Wick et al. [35]	IWDAS'22	> 0	3.96	12.20	-	-
Li et al. [19]	AAAI'23	700	2.89	-	-	-
Kizilirmak et al. [18]	'23	2.5	5.05	14.46	-	-
DDViT (+V)			3.34	11.11	3.37	11.09

4.3 Ablation Experiments

Masking Baseline models utilising MP, MADS, and convolution downsampling layer were trained. The convolution downsampling layer used the same kernel size and stride as the MP layer. For each type of downsampling, 2 models were trained where one was trained with masking at all multi-head attention (MHA) layers while the other without.

Results are shown in Tab. 3 where masking improved the CER of both MP and MADS baseline models. This was especially prominent in 'Baseline MADS', where overfitting was observed in the unmasked model with about 29% CER in validation set. The generalisation greatly improved where the CER dropped to about 2.98% when mask was applied to MHA. For 'Baseline MP', the masking at MHA improved the CER by about 0.39% points for validation set. However,

Table 3. Results of baseline models trained with and without masking applied. 'Unmasked MHA' represents model trained without masking multi-head attention (MHA). 'Masked MHA' represents mask applied in all MHA.

Method	Validation	
	CER (%)	WER (%)
Baseline MP		
Unmasked MHA	3.18	11.14
Masked MHA	2.79	9.87
Baseline MADS		
Unmasked MHA	29.00	63.68
Masked MHA	2.98	10.55
Baseline Conv		
Unmasked MHA	3.29	11.72
Masked MHA	3.37	11.77

result did not improve when masking was applied on the MHA for 'Baseline Conv' where the CER increased from 3.29% to 3.37%.

Ensemble VS. Dual Downsample Vision Transformer DDViT and baseline models with convolution, MP, and MADS downsampling layers were trained using the same hyperparameters and settings. The convolution layer matched MP in kernel size and stride. Baseline models were also ensembled, with final predictions selected via the proposed dual maps scoring method. As shown in Tab. 4, wall-clock inference time was measured with batch size 1 on an RTX 3090. Among single models, MP achieved the best CER (2.79%), followed by MADS (2.98%) and convolution (3.29%). The ensemble of MP and MADS ('Ensem. MP & MADS') improved CER to 2.62%, but doubled both parameters and inference time. Including convolution in the ensemble ('Ensem. Conv & MP & MADS') worsened performance (2.91% CER), likely due to the weaker "Baseline Conv". In contrast, DDViT matched 'Ensem. MP & MADS' (2.63% CER) while using only 51% of the parameters and 62% of the inference time, demonstrating its superior efficiency.

Table 4. Comparison of ensemble of baseline models with DDViT.

Method	# Params. (mil.)	Time (ms)	Validation CER (%)	Validation WER (%)
Baseline Conv	16.6	12.8	3.29	11.72
Baseline MP	14.5	12.2	2.79	9.87
Baseline MADS	14.6	12.4	2.98	10.55
Ensem. MP & MADS	29.1	24.9	2.62	9.14
Ensem. Conv & MP & MADS	45.7	37.9	2.91	10.21
DDViT	14.8	15.4	2.63	9.29

Scoring Methods The accuracy of different scoring methods (as introduced in Sec. 3.4) on the ensemble of baseline models and DDViT are tabulated in Tab. 5. Generally, the 'Dual Maps' scoring method gave the best CER. For the ensemble

of 3 baseline models, 'Ensem. Conv& MP & MADS', the 'Best Path' method gave the next best CER. However, for 'Ensem. MP & MADS' and DDViT, the 'String' method gave the next best CER followed by 'Best Path' method.

Table 5. Results of various scoring methods.

Method	**Best Path** CER (%)	**String** CER (%)	**Dual Maps** CER (%)
Ensem. Conv & MP & MADS	4.74	10.99	2.91
Ensem. MP & MADS	2.89	2.69	2.62
DDViT	2.92	2.80	2.63

Comparison of Latency The wall-clock inference time of publicly available works are tabulated in Tab. 6. Batch size of 1 was used with an RTX 3090. The results of proposed models were comparable with the work by Li et al. [19] which used over 334 million parameters. As compared to it, 'DDViT (+V)' achieved a CER of 3.34% which is 0.08% points lower. This was achieved with 4.4% number of parameters (at 14.8 million) and 12.6% inference time (at 15.4ms). Furthermore, 'DDViT (+V)' achieved at least 0.2% points improvement in CER over 'Baseline MP (+V)' with just slightly more parameters and inference time.

Table 6. Comparison of wall-clock time per image during inference with TrOCR variants (Small, Base, and Large).

Method	**# Params.** (mil.)	**Time** (ms)	**IAM-A** CER (%)	**IAM-B** CER (%)
Coquenet et al. (+P) [6]	2.7	6.52	-	4.54
Li et al. ($\text{TrOCR}_{\text{SMALL}}$) [19]	62	41.9	4.22	-
Li et al. ($\text{TrOCR}_{\text{BASE}}$) [19]	334	122	3.42	-
Li et al. ($\text{TrOCR}_{\text{LARGE}}$) [19]	558	184	2.89	-
Baseline MP (+V)	14.5	12.2	3.66	3.59
DDViT (+V)	14.8	15.4	3.34	3.37

5 Conclusion

This work introduced a vision transformer baseline and the Dual Downsample Vision Transformer (DDViT), enhanced by a dual maps scoring method, for line-level HTR on small open-source datasets. DDViT achieves state-of-the-art CERs (e.g., 3.34% on IAM-A) with minimal increases in parameters and inference time, outperforming prior works without requiring additional data. These results highlight the potential of efficient transformer-based models in resource-constrained HTR tasks. Future work could explore DDViT's applicability to multilingual datasets or real-time applications.

References

1. Al Azawi, M., Liwicki, M., Breuel, T.M.: Combination of multiple aligned recognition outputs using wfst and lstm. In: Proceedings of International Conference on Document Analysis and Recognition. pp. 31–35. IEEE (2015)
2. Augustin, E., Carré, M., Grosicki, E., Brodin, J.M., Geoffrois, E., Prêteux, F.: Rimes evaluation campaign for handwritten mail processing. In: Proceedings of International Workshop on Frontiers in Handwriting Recognition. pp. 231–235 (2006)
3. Bluche, T.: Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. Advances in Neural Information Processing Systems **29** (2016)
4. Bluche, T., Louradour, J., Messina, R.: Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention. In: Proceedings ofInternational Conference on Document Analysis and Recognition. vol. 1, pp. 1050–1055. IEEE (2017)
5. Cascianelli, S., Pippi, V., Maarand, M., Cornia, M., Baraldi, L., Kermorvant, C., Cucchiara, R.: The lam dataset: A novel benchmark for line-level handwritten text recognition. In: Proceedings of International Conference on Pattern Recognition. pp. 1506–1513. IEEE (2022)
6. 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)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). pp. 4171–4186 (2019)
8. Diaz, D.H., Qin, S., Ingle, R., Fujii, Y., Bissacco, A.: Rethinking text line recognition models. arXiv preprint arXiv:2104.07787 (2021)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of International Conference on Learning Representations (2021)
10. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of International Conference on Machine Learning. pp. 369–376 (2006)
11. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(5), 855–868 (2008)
12. Gui, L., Liang, X., Chang, X., Hauptmann, A.G.: Adaptive context-aware reinforced agent for handwritten text recognition. In: Proceedings of British Machine Vision Conference. vol. 207 (2018)
13. Hamdan, M., Chaudhary, H., Bali, A., Cheriet, M.: Refocus attention span networks for handwriting line recognition. International Journal on Document Analysis and Recognition **26**(2), 131–147 (2023)
14. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: Proceedings of European Conference on Computer Vision. pp. 646–661. Springer (2016)

15. 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)
16. Kang, L., Riba, P., Villegas, M., Fornés, A., Rusiñol, M.: Candidate fusion: Integrating language modelling into a sequence-to-sequence handwritten word recognition architecture. Pattern Recognition **112**, 107790 (2021)
17. Kang, L., Toledo, J.I., Riba, P., Villegas, M., Fornés, A., Rusinol, M.: Convolve, attend and spell: An attention-based sequence-to-sequence model for handwritten word recognition. In: Proceedings of German Conference on Pattern Recognition. pp. 459–472. Springer (2018)
18. Kizilirmak, F., Yanikoglu, B.: Cnn-bilstm model for english handwriting recognition: Comprehensive evaluation on the iam dataset. arXiv preprint arXiv:2307.00664 (2023)
19. Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: Trocr: Transformer-based optical character recognition with pre-trained models. In: Proceedings of AAAI Conference on Artificial Intelligence. vol. 37, pp. 13094–13102 (2023)
20. Li, Y., Chen, D., Tang, T., Shen, X.: Htr-vt: Handwritten text recognition with vision transformer. Pattern Recognition **158**, 110967 (2025)
21. Lyu, P., Zhang, C., Liu, S., Qiao, M., Xu, Y., Wu, L., Yao, K., Han, J., Ding, E., Wang, J.: Maskocr: Scene text recognition with masked vision-language pre-training. Transactions on Machine Learning Research (2024)
22. 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)
23. Michael, J., Labahn, R., Grüning, T., Zöllner, J.: Evaluating sequence-to-sequence models for handwritten text recognition. In: Proceedings of International Conference on Document Analysis and Recognition. pp. 1286–1293. IEEE (2019)
24. Poulos, J., Valle, R.: Attention networks for image-to-text. ArXiv **abs/1712.04046** (2017), https://api.semanticscholar.org/CorpusID:29748509
25. Poulos, J., Valle, R.: Character-based handwritten text transcription with attention networks. Neural Computing and Applications **33**(16), 10563–10573 (2021)
26. Puigcerver, J.: Are multidimensional recurrent layers really necessary for handwritten text recognition? In: Proceedings of International Conference on Document Analysis and Recognition. vol. 1, pp. 67–72. IEEE (2017)
27. Reul, C., Springmann, U., Wick, C., Puppe, F.: Improving ocr accuracy on early printed books by utilizing cross fold training and voting. In: Proceedings of International Workshop on Document Analysis Systems. pp. 423–428. IEEE (2018)
28. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018)
29. Tan, Y.L., Kong, A.W.K., Kim, J.J.: Pure transformer with integrated experts for scene text recognition. In: Proceedings of European Conference on Computer Vision. pp. 481–497. Springer (2022)
30. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. In: Proceedings of International Conference on Computer Vision. pp. 32–42 (2021)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Proceedings of Advances in Neural Information Processing Systems. vol. 30 (2017)

32. Wang, T., Zhu, Y., Jin, L., Luo, C., Chen, X., Wu, Y., Wang, Q., Cai, M.: Decoupled attention network for text recognition. In: Proceedings of AAAI Conference on Artificial Intelligence. vol. 34, pp. 12216–12224 (April 2020). https://doi.org/10.1609/aaai.v34i07.6903, https://ojs.aaai.org/index.php/AAAI/article/view/6903
33. Wemhoener, D., Yalniz, I.Z., Manmatha, R.: Creating an improved version using noisy ocr from multiple editions. In: Proceedings of International Conference on Document Analysis and Recognition. pp. 160–164. IEEE (2013)
34. 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)
35. Wick, C., Zöllner, J., Grüning, T.: Rescoring sequence-to-sequence models for text line recognition with ctc-prefixes. In: Proceedings of International Workshop on Document Analysis Systems. pp. 260–274. Springer (2022)
36. Yousef, M., Hussain, K.F., Mohammed, U.S.: Accurate, data-efficient, unconstrained text recognition with convolutional neural networks. Pattern Recognition **108**, 107482 (2020)
37. Zaheer, M., Guruganesh, G., Dubey, K.A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al.: Big bird: Transformers for longer sequences. Advances in Neural Information Processing Systems **33**, 17283–17297 (2020)