IEEE Transactions on Image Processing

HTR-JAND: Handwritten Text Recognition with Joint Attention Network and Knowledge Distillation

Journal:	*Transactions on Image Processing*
Manuscript ID	TIP-33699-2024
Manuscript Type:	Regular Paper (S1)
Date Submitted by the Author:	01-Nov-2024
Complete List of Authors:	Hamdan, Mohammed; Ecole de technologie superieure, Systems Engineerning Rahiche, Abderrahmane Cheriet, Mohamed; Ecole de Technologie Superieure, Lab for Imagery, Vision, and Artificial Intelligence
Subject Category Please select at least one subject category that best reflects the scope of your manuscript:	Image & Video Processing Techniques, Image and Video Analysis, Synthesis and Retrieval
EDICS:	22. ELI-DOC Scanned Document Analysis, Processing, & Coding < Electronic Imaging, 36. ARS-SRV Image and Video Synthesis, Rendering, and Visualization < Image and Video Analysis, Synthesis and Retrieval, 35. ARS-SRE Image and Video Storage and Retrieval < Image and Video Analysis, Synthesis and Retrieval, 33. ARS-IIU Image and Video Interpretation and Understanding < Image and Video Analysis, Synthesis and Retrieval

Author's Responses to Custom Submission Questions

If the manuscript has more than one author, briefly describe every author's significant intellectual contribution (max 25 words per author). If this does not apply, type N/A.	- Mohamed Hamdan: Primary architect of HTR-JAND framework, implementation of knowledge distillation, Development of attention mechanisms, experimental design, results analysis, and manuscript preparation. - Abderrahmane Rahiche: Theoretical framework development - Mohamed Cheriet: Manuscript review
Is this manuscript a resubmission of, or related to, a previously rejected manuscript?	This manuscript presents original work and has not been published or submitted elsewhere. All related literature is properly cited.
If "Yes", specify the publication venue and manuscript ID of the previous submission and upload a supporting document detailing how the resubmission has addressed the concerns raised during the previous review. If this does not apply, type N/A.	CUST_RESUBMISSION_DETAILS :No data available.
Is this manuscript an extended version of a conference publication?	No, this is not a resubmission of a previously rejected manuscript.
If "Yes", provide the full citation of the conference submission or publication. If this does not apply, type N/A.	CUST_CONFERENCE_PUBLICATION_DETAILS :No data available.
Is this manuscript related to any other papers of the authors that are either published, accepted for publication, or currently under review, and that are not included among the references cited in the manuscript?	No, this is not an extended version of a conference publication.
If "Yes", please list these papers below. Except for permitted preprints, explain why these papers are not included among the references cited in the manuscript and how they are different from the manuscript. Include any unpublished papers as "Supporting Documents". If this does not apply, type N/A.	CUST_RELATED_PAPER_DETAILS :No data available.
What is the contribution of this paper, within the scope of Transactions on Image Processing?	This work falls directly within TIP's scope as it presents fundamental algorithmic contributions to image processing through: - Novel architectural components for visual feature extraction - Advanced attention mechanisms for image sequence processing - Comprehensive evaluation on image-based text

	recognition - Broad applicability to document image analysis
Why is the contribution significant (What impact will it have)?	CUST_SIGNIFICANCE :No data available.
What are the three papers in the published literature most closely related to this paper? Please provide full citation details, including DOI references where possible.	1. Retsinas, George, et al. "Best practices for a handwritten text recognition system." International Workshop on Document Analysis Systems. Cham: Springer International Publishing, 2022. 2. Yousef, Mohamed, Khaled F. Hussain, and Usama S. Mohammed. "Accurate, data-efficient, unconstrained text recognition with convolutional neural networks." Pattern Recognition 108 (2020): 107482. 3. Tassopoulou, Vasiliki, George Retsinas, and Petros Maragos. "Enhancing handwritten text recognition with n-gram sequence decomposition and multitask learning." 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021.
What is distinctive/new about the current paper relative to these previously published works?	Distinctive Contributions: - First to combine knowledge distillation with joint attention for HTR - Superior performance while achieving 48% parameter reduction - Novel integration of curriculum learning with synthetic data generation - State-of-the-art results across multiple benchmark datasets

HTR-JAND: Handwritten Text Recognition with Joint Attention Network and Knowledge Distillation

Mohammed Hamdan, Abderrahmane Rahiche, *Member, IEEE,* Mohamed Cheriet, *Senior Member, IEEE,*

***Abstract*—The digitization and accurate recognition of handwritten historical documents remain crucial for preserving cultural heritage and making historical archives accessible to researchers and the public. Despite significant advances in deep learning, current Handwritten Text Recognition (HTR) systems struggle with the inherent complexity of historical documents, including diverse writing styles, degraded text quality, and computational efficiency requirements across multiple languages and time periods. This paper introduces HTR-JAND (HTR-JAND: Handwritten Text Recognition with Joint Attention Network and Knowledge Distillation), an efficient HTR framework that combines advanced feature extraction with knowledge distillation. Our architecture incorporates three key components: (1) a CNN architecture integrating FullGatedConv2d layers with Squeeze-and-Excitation blocks for adaptive feature extraction, (2) a Combined Attention mechanism fusing Multi-Head Self-Attention with Proxima Attention for robust sequence modeling, and (3) a Knowledge Distillation framework enabling efficient model compression while preserving accuracy through curriculum-based training. The HTR-JAND framework implements a multi-stage training approach combining curriculum learning, synthetic data generation, and multi-task learning for cross-dataset knowledge transfer. We enhance recognition accuracy through context-aware T5 post-processing, particularly effective for historical documents. Comprehensive evaluations demonstrate HTR-JAND's effectiveness, achieving state-of-the-art Character Error Rates (CER) of 1.23%, 1.02%, and 2.02% on IAM, RIMES, and Bentham datasets respectively. Our Student model achieves a 48% parameter reduction (0.75M versus 1.5M parameters) while maintaining competitive performance through efficient knowledge transfer. Source code and pre-trained models are available at Github.**

***Index Terms*—Handwritten text recognition, knowledge distillation, attention mechanisms, Multihead attention, Proxima attention, multi-task learning, curriculum learning, T5 postprocessing.**

I. Introduction

HANDWRITTEN text recognition in historical documents represents a cornerstone of digital humanities and cultural heritage preservation. The ability to accurately convert handwritten documents into machine-readable text is essential for making centuries of historical records, manuscripts, and cultural artifacts accessible to researchers, historians, and the public. This task presents significant challenges due to writing style variability, document degradation, and diverse linguistic content across multiple time periods [1], [2]. Figure 1 illustrates these challenges through representative samples from different historical periods and writing styles, highlighting the complexity of developing robust recognition systems.

Authors are with the Synchromedia laboratory, École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, Canada.

Manuscript received October XX, 2024; revised XX XX, 2025.

Traditional approaches based on segmentation methods [3], [4] and complex processing pipelines [5], [6] struggle with capturing the nuanced relationships between handwriting styles and textual content. These methods often require extensive preprocessing and manual intervention, limiting their applicability in large-scale digitization projects. Current deep learning methods, while promising, face three fundamental limitations: inconsistent generalization across writing styles and historical periods [7], [8], difficulties in handling long text sequences [9], [10], and computational requirements that restrict practical deployment [11], [12]. While attention mechanisms have improved sequence modeling capabilities [13], [14], existing approaches continue to struggle with balancing recognition accuracy and computational efficiency.

These challenges are further compounded by the lack of robust mechanisms for handling historical character variations and archaic writing styles [1]. Combined with the computational demands of processing handwritten text recognition [15], [16], these limitations highlight the need for an integrated approach that addresses both accuracy and efficiency requirements.

To address these challenges, we present HTR-JAND (HTR-JAND: Handwritten Text Recognition with Joint Attention Network and Knowledge Distillation), an end-to-end framework that combines efficient feature extraction with knowledge transfer capabilities. Our approach includes several key components:

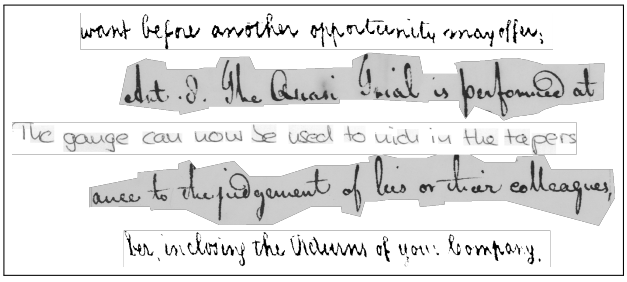

Fig. 1: Sample images from different datasets, demonstrating the range of challenges including writing style variability, non-standard character shapes, and contextual dependencies.

- A comprehensive preprocessing pipeline that combines character set unification across datasets with adaptive oversampling, achieving balanced representation while maintaining a unified vocabulary of 103 characters across diverse historical periods and writing styles.
- A CNN architecture combining FullGatedConv2d layers with Squeeze-and-Excitation blocks for adaptive feature

extraction, inspired by recent advances in visual recognition [17].
- A Combined Attention mechanism that integrates Multi-Head Self-Attention with Proxima Attention, building upon successful approaches in sequence modeling [8], [13].
- A knowledge distillation framework enabling compact model deployment while maintaining performance, extending techniques for model compression [18].
- Training strategies incorporating curriculum learning with synthetic data generation [19], ensemble learning, and multi-task learning.
- Context-aware post-processing using a fine-tuned T5 model to improve recognition accuracy in historical texts.
- Comprehensive evaluations on standard benchmarks described in subsection III-A demonstrate HTR-JAND's effectiveness. The framework achieves state-of-the-art Character Error Rates of 1.23%, 1.02%, and 2.02% on IAM, RIMES, and Bentham datasets respectively, while maintaining practical efficiency through significant parameter reduction.

The paper is structured as follows: Section II reviews recent HTR developments; Section III details the model architecture and loss function design; Section IV describes training strategies; Section VI presents experimental results; and Section VII concludes the paper with findings and future directions.

II. Related Work

Handwritten Text Recognition (HTR) has seen significant advancements with deep learning techniques, and this section offers an overview of key developments by showcasing architectural innovations and identifying gaps our work addresses; Table I summarizes key studies focusing on architectural innovations, attention mechanisms, and performance on benchmark datasets, with notes defining abbreviations: GC (Gated Convolution), SE (Squeeze-and-Excitation Blocks), CA (Combined Attention), KD (Knowledge Distillation), CL (Curriculum Learning), AR (Aspect Ratio Preservation), and PP (Post-processing).

TABLE I: Overview of studies showing different architectural components implemented

Study	GC	SE	CA	KD	CL	AR	PP
Graves et al. [20]	✓						
Puigcerver [10]							
Bluche [2]	✓						
Chowdhury et al. [8]			✓				
Kang et al. [13]			✓				
Wigington et al. [19]					✓		
Hamdan et al. [15]		✓					
Flor et al. [17]	✓	✓					✓
Retsinas et al. [21]						✓	
(HTR-JAND) this Work	✓	✓	✓	✓	✓	✓	✓

A. Architectural Evolution in HTR

The foundation of modern HTR systems was laid by traditional Hidden Markov Models (HMM) [4], [6], [22], which provided probabilistic frameworks for sequence modeling but struggled with long-range dependencies and required careful feature engineering. This was followed by Graves et al. [20] introducing Connectionist Temporal Classification (CTC), enabling end-to-end training on unsegmented sequence data. This work, utilizing Bidirectional Long Short-Term Memory (BLSTM) networks, marked a significant departure from traditional HMM approaches by allowing the model to learn feature representations directly from raw input data.

Subsequent research focused on sequence-to-sequence modeling [23]–[25], which treated HTR as a translation problem from image to text, and integrating Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs). These approaches enabled more flexible handling of variable-length inputs and outputs while capturing both spatial and temporal dependencies. Puigcerver [10] demonstrated the effectiveness of CNN-LSTM architectures combined with CTC loss, achieving competitive results on standard benchmarks. This approach set a new baseline for HTR systems, balancing feature extraction capabilities with sequential modeling.

Further architectural innovations emerged to address specific challenges in HTR. Bluche [2] introduced gated convolutional layers to better handle long text sequences, while Dutta et al. [5] employed Spatial Transformer Networks to address geometric distortions in handwritten text. However, the challenge of handwriting variability, especially in historical documents, continues to impact model generalization [1].

B. Attention Mechanisms and Advanced Techniques

Attention mechanisms [26]–[29] have become increasingly prominent in HTR, allowing models to focus on relevant parts of the input during recognition. These mechanisms dynamically weight different regions of the input based on their relevance to the current prediction, enabling more precise character recognition and better handling of complex layouts. Self-attention approaches [30], [31] further enhanced this capability by calculating responses at particular sequence locations by attending to the entire sequence, effectively capturing global dependencies without the need for recurrent connections. Chowdhury et al. [8] and Kang et al. [13] demonstrated the effectiveness of attention in end-to-end neural models and Transformer architectures, respectively, though capturing long-range dependencies in very long text sequences remains challenging [13].

Recent works have explored more sophisticated techniques to improve HTR performance. Data augmentation strategies [32], [33] have proven effective for handling limited data scenarios, incorporating techniques such as elastic distortions, affine transformations, and synthetic data generation to improve model robustness. These methods have been particularly valuable for historical document recognition where training data is scarce. Hamdan et al. [15] incorporated Squeeze-and-Excitation (SE) blocks to enhance feature representation, while Flor et al. [17] combined gated convolutions with SE blocks and introduced post-processing techniques. Retsinas et al. [21] focused on preserving aspect ratios of input images, addressing the issue of information loss during preprocessing.

C. Efficiency and Learning Strategies

As HTR models grew in complexity, research began to focus on improving efficiency and generalization. Wigington et al. [19] highlighted the importance of data augmentation and curriculum learning strategies. Knowledge distillation, as demonstrated by You et al. [18], emerged as an effective technique for transferring knowledge from large teacher models to smaller, more efficient student models. However, balancing computational efficiency with recognition accuracy, particularly for deployment in resource-constrained environments, remains an ongoing concern [34], [35], and addressing data scarcity for historical or less common languages continues to challenge the field [7].

As shown in Table I, existing approaches have typically focused on individual components or techniques in isolation. Our work uniquely integrates multiple state-of-the-art techniques while introducing new elements. We combine gated convolutions with SE blocks for enhanced feature extraction, integrate a novel combined attention mechanism for improved handling of long-range dependencies, and implement knowledge distillation alongside curriculum learning strategies for better efficiency and generalization. The preservation of aspect ratios and advanced post-processing techniques further enhance our model's ability to handle diverse handwriting styles and complex documents. This comprehensive approach represents a significant step toward more robust and efficient HTR systems, addressing multiple challenges concurrently rather than in isolation.

III. Methodology

Our approach to Handwritten Text Recognition (HTR) introduces several key innovations to address the challenges of diverse writing styles, historical documents, and computational efficiency. This section details our methodological contributions, emphasizing the novel aspects of our architecture and training strategy.

A. Data Preprocessing and Augmentation

To facilitate knowledge distillation and standardize training across multiple datasets including including IAM [36], RIMES [37], Bentham [38], Saint Gall [1], and Washington [7], our preprocessing approach begins with character set unification. The process removes infrequent characters that would not significantly impact classifier performance, resulting in a unified set of 103 unique characters across all datasets. As shown in Figure 2, character frequencies exhibit considerable variation, with some characters appearing frequently (lowercase letters and spaces) while others occur rarely.

Table II presents key statistics for each dataset after preprocessing, including a buffer of 2 added to the maximum sequence length to accommodate variations during inference.

Our preprocessing pipeline addresses three key challenges: handwriting style variability, limited labeled data availability, and temporal coherence preservation. Each input image $\boldsymbol{I}$ undergoes normalization to a standard size of 68×864 pixels:

$$\boldsymbol{I}'_{x,y} = 2 \cdot \frac{\boldsymbol{I}_{x,y} - \min(\boldsymbol{I})}{\max(\boldsymbol{I}) - \min(\boldsymbol{I})} - 1. \tag{1}$$

TABLE II: Dataset statistics after preprocessing

Dataset	Train	Valid	Test	Vocab	Max Len +2 Buffer
IAM	6,161	900	1,861	79	93
RIMES	10,193	1,133	778	100	110
Washington	325	168	163	68	61
Bentham	9,195	1,415	860	94	103
Saint Gall	468	235	707	48	74
Combined	26,342	3,851	4,369	103	110

Algorithm 1 Preprocessing Pipeline with Synthetic Data (PPS)

Input: D, C, F, r, α
Output: D'
1: $\mathbf{D}_n \leftarrow \text{Normalize}(D)$ {Eq. 1}
2: $\mathbf{D}_a \leftarrow \text{Augment}(\mathbf{D}_n)$ {Apply transforms}
3: $\mathbf{D}_s \leftarrow \text{GenerateSynthetic}(C, F, r)$ {Algo 2}
4: $\mathbf{D}_t \leftarrow \text{Tokenize}(\mathbf{D}_a \cup \mathbf{D}_s, C)$
5: $\mathbf{D}' \leftarrow \text{BalanceClasses}(\mathbf{D}_t, \alpha)$
6: **return** $\mathbf{D}'$

The complete preprocessing workflow follows Algorithm 1:

Data augmentation applies transformations $\boldsymbol{T} = \{t_1, \ldots, t_n\}$ to each image:

$$\boldsymbol{I}_{\text{aug}} = t_n(\ldots t_2(t_1(\boldsymbol{I}))). \tag{2}$$

The synthetic data generation process is defined in Algorithm 2:

The pipeline incorporates three key strategies for training stability: curriculum-based synthetic ratio adjustment, performance-based adaptive synthetic data integration with a 10% initial ratio, and enhanced augmentation techniques.

For class balancing, we implement adaptive oversampling:

$$w_c = \max(1, \frac{\bar{f}}{\epsilon + f_c}), \tag{3}$$

where w_c represents the sampling weight for character c, f_c is its frequency, $\bar{f}$ denotes mean character frequency, and ϵ prevents division by zero.

This comprehensive approach ensures effective preprocessing across our diverse dataset while maintaining consistency and stability in the training process.

B. The Proposed Model

The proposed HTR architecture addresses handwritten text recognition challenges through a hierarchical structure combining convolutional neural networks, recurrent layers, and attention mechanisms. As shown in Figure 3, the architecture processes input text line images through an encoder-decoder pipeline, employing a Teacher-Student framework as described in the next subsection III-C to balance recognition accuracy with computational efficiency.

1) Architecture Overview: The model employs a Teacher-Student framework where the Teacher model provides a comprehensive architecture that is later distilled into a more efficient Student model. The Teacher model integrates five key components, as illustrated in Figure 3: CNN blocks with

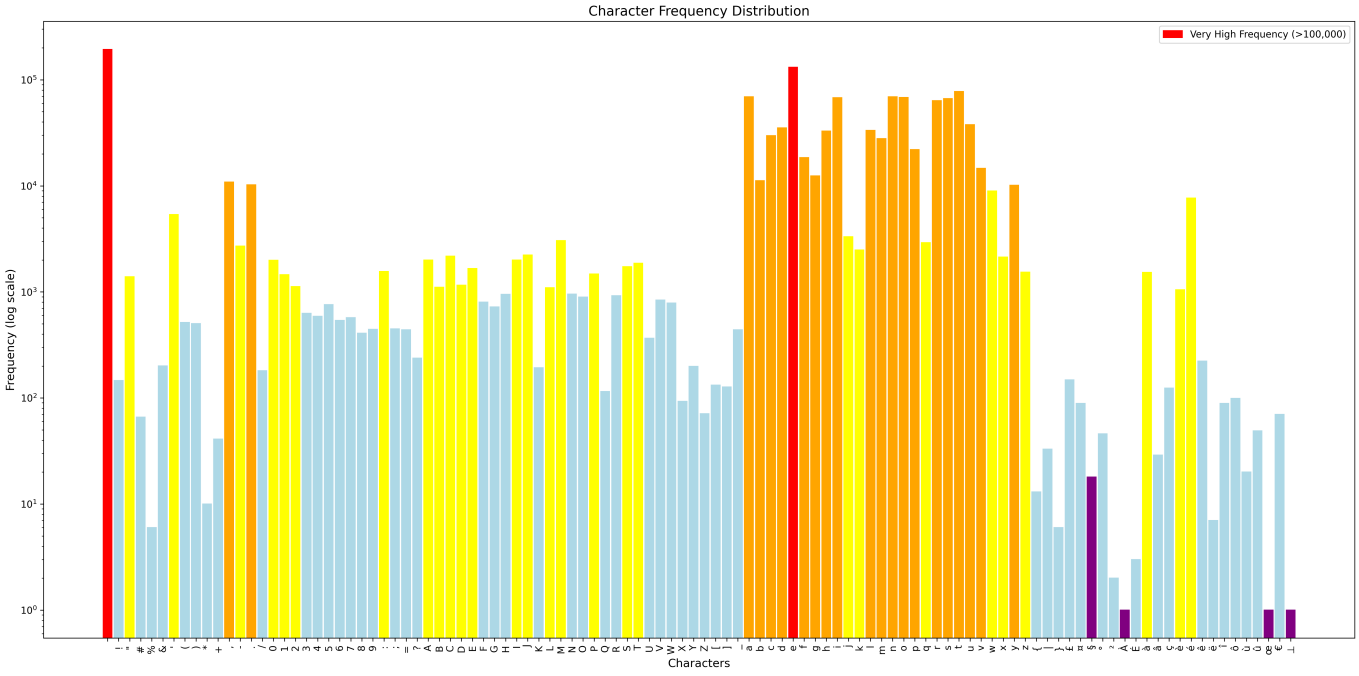

Fig. 2: Distribution of character frequencies across the combined datasets. Note the removal of infrequent characters such as '§', 'À', and 'òe'.

Algorithm 2 Synthetic Data Generation (SDG)

Input: C, F, r, D
Output: $\mathbf{D}_s$

```
n ← |D| · r/(1 − r)
for i = 1 to n do
    t ← RandomText(C)
    f ← RandomChoice(F)
    I ← RenderText(t, f)
    I_aug ← Augment(I)
    D_s ← D_s ∪ {(I_aug, t)}
end for
return D_s
```

Squeeze-and-Excitation (SE) modules, FullGatedConv2d layers for adaptive feature extraction, bidirectional LSTM layers for sequence modeling, Multi-Head Self-Attention combined with Proxima Attention, and CTC-based decoding with auxiliary classification.

The model processes grayscale input images of size 68×864 through progressive feature extraction stages.

2) CNN Feature Extraction: The CNN backbone combines FullGatedConv2d layers with SE modules. Each CNN block executes operations according to:

$$\boldsymbol{f}_l = \text{SE}(\text{MaxPool}(\text{ReLU}(\text{BN}(\boldsymbol{W}_l * \boldsymbol{f}_{l-1} + \mathbf{b}_l)))), \quad (4)$$

where $\boldsymbol{f}_l \in \mathbb{R}^{C_l \times H_l \times W_l}$ represents the feature map at layer l, $\boldsymbol{W}_l$ and $\mathbf{b}_l$ denote convolutional parameters, and $*$ indicates convolution. The SE operation adaptively recalibrates channel responses:

$$\boldsymbol{f}_{\text{SE}} = \boldsymbol{f}_l \cdot \sigma(\boldsymbol{W}_2\text{ReLU}(\boldsymbol{W}_1\text{GAP}(\boldsymbol{f}_l))). \quad (5)$$

The FullGatedConv2d layer implements an adaptive gating mechanism:

$$\text{FullGatedConv2d}(\boldsymbol{X}) = (\boldsymbol{W}_1 * \boldsymbol{X}) \odot \sigma(\boldsymbol{W}_2 * \boldsymbol{X}). \quad (6)$$

The network employs a strategic pooling approach to maintain sequence information:

$$\text{MaxPool}_{2,1}(\boldsymbol{X})_{i,j} = \max_{0\le m<2} (X_{2i+m,j}). \quad (7)$$

3) Sequence Modeling with BiLSTM: The CNN features feed into four bidirectional LSTM layers for temporal modeling:

$$\mathbf{h}_t = [\overrightarrow{\text{LSTM}}(\boldsymbol{X}_t, \overrightarrow{\mathbf{h}}_{t-1}); \overleftarrow{\text{LSTM}}(\boldsymbol{X}_t, \overleftarrow{\mathbf{h}}_{t+1})], \quad (8)$$

where, each LSTM cell follows:

$$\boldsymbol{I}_t = \sigma(\boldsymbol{W}_i[\mathbf{h}_{t-1}, \boldsymbol{X}_t] + \mathbf{b}_i) \quad (9)$$
$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f[\mathbf{h}_{t-1}, \boldsymbol{X}_t] + \mathbf{b}_f) \quad (10)$$
$$\mathbf{o}_t = \sigma(\boldsymbol{W}_o[\mathbf{h}_{t-1}, \boldsymbol{X}_t] + \mathbf{b}_o) \quad (11)$$
$$\tilde{\mathbf{c}}_t = \tanh(\boldsymbol{W}_c[\mathbf{h}_{t-1}, \boldsymbol{X}_t] + \mathbf{b}_c) \quad (12)$$
$$\mathbf{c}_t = \boldsymbol{f}_t \odot \mathbf{c}_{t-1} + \boldsymbol{I}_t \odot \tilde{\mathbf{c}}_t \quad (13)$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \quad (14)$$

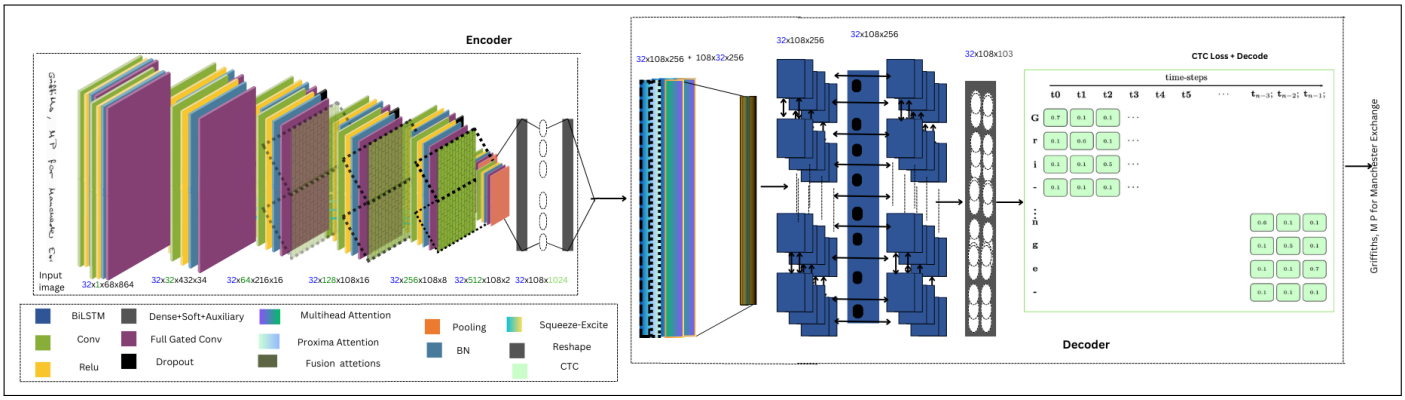

Fig. 3: Proposed HTR Model Architecture: Data flow through CNN feature extraction, LSTM sequence modeling, and Combined Attention mechanisms. Additionally, CTC Matrix for "Griffiths, M P for Manchester Exchange" showing probabilities for first "Gri-" and last "-nge" ('-' represents blank symbol for CTC alignment).

4) Combined Attention Mechanism: The model integrates Multi-Head Self-Attention with Proxima Attention. The base attention operation computes:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V}. \tag{15}$$

Multi-Head Attention extends this through parallel attention operations:

$$\text{MultiHead}(\boldsymbol{X}) = \text{Concat}(\text{head}_1, ..., \text{head}_h)\boldsymbol{W}^O \tag{16}$$

Proxima Attention commuted using Eq. 16 while introducing dynamic query updates:

$$\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K, \quad \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V \tag{17}$$

The combined attention output is:

$$\mathbf{O}_{\text{combined}} = \text{LayerNorm}(\boldsymbol{W}_f[\mathbf{O}_{\text{MHA}}; \mathbf{O}_{\text{Proxima}}] + \boldsymbol{X}) \tag{18}$$

5) Student Model Architecture: The Student model maintains the architectural principles while reducing complexity through: - Three CNN blocks instead of five - Channel dimensions starting at 16 instead of 32 - One attention head instead of two - Reduced hidden dimensions in LSTM layers to 64 instead of 128.

This design achieves a 48% parameter reduction (750,654 parameters versus 1,504,544) while preserving recognition capabilities through knowledge distillation.

C. Knowledge Distillation

Our knowledge distillation approach enables efficient model deployment by transferring learned representations from a high-capacity Teacher model to a compact Student model. As illustrated in Figure 4, the framework employs a Teacher model with full capacity (1.5M parameters) to guide the training of a more efficient Student model (0.75M parameters), addressing the practical challenges of deploying complex HTR systems in resource-constrained environments while maintaining recognition accuracy.

The knowledge transfer process, visualized in the right portion of Figure 4, shows how information flows from the Teacher to the Student model through multiple loss components. This design allows the Student to learn not only from ground truth labels but also from the Teacher's learned representations and confidence scores, particularly beneficial for challenging cases and rare characters in historical documents.

1) Multi-Component Loss Framework: The knowledge transfer process is guided by a comprehensive loss function that combines four complementary components, each serving a specific purpose in the training process:

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{ctc}} + \beta\mathcal{L}_{\text{ce}} + \gamma\mathcal{L}_{\text{kd}} + \delta\mathcal{L}_{\text{aux}}, \tag{19}$$

where α, β, γ, and δ are balancing hyperparameters dynamically adjusted during training to control the contribution of each loss component. As shown in Figure 4, these components work together to ensure effective knowledge transfer while maintaining recognition accuracy.

The CTC loss ($\mathcal{L}_{\text{ctc}}$) addresses the fundamental sequence alignment challenge in HTR, handling variable-length inputs without requiring explicit alignments:

$$p(\mathbf{y}|\boldsymbol{X}) = \sum_{\boldsymbol{\pi}\in\mathcal{B}^{-1}(\mathbf{y})} p(\boldsymbol{\pi}|\boldsymbol{X}), \tag{20}$$

$$\mathcal{L}_{\text{ctc}} = -\log(p(\mathbf{y}|\boldsymbol{X})), \tag{21}$$

where $\boldsymbol{\pi}$ represents possible alignments between input and output sequences, including blank tokens for flexible alignment.

The cross-entropy loss ($\mathcal{L}_{\text{ce}}$) provides direct character-level supervision, particularly important for maintaining accuracy on individual character recognition:

$$\mathcal{L}_{\text{ce}} = -\sum_i y_i \log(\hat{y}_i). \tag{22}$$

The knowledge distillation loss ($\mathcal{L}_{\text{kd}}$), central to our framework as depicted in Figure 4, facilitates the transfer of learned representations from Teacher to Student:

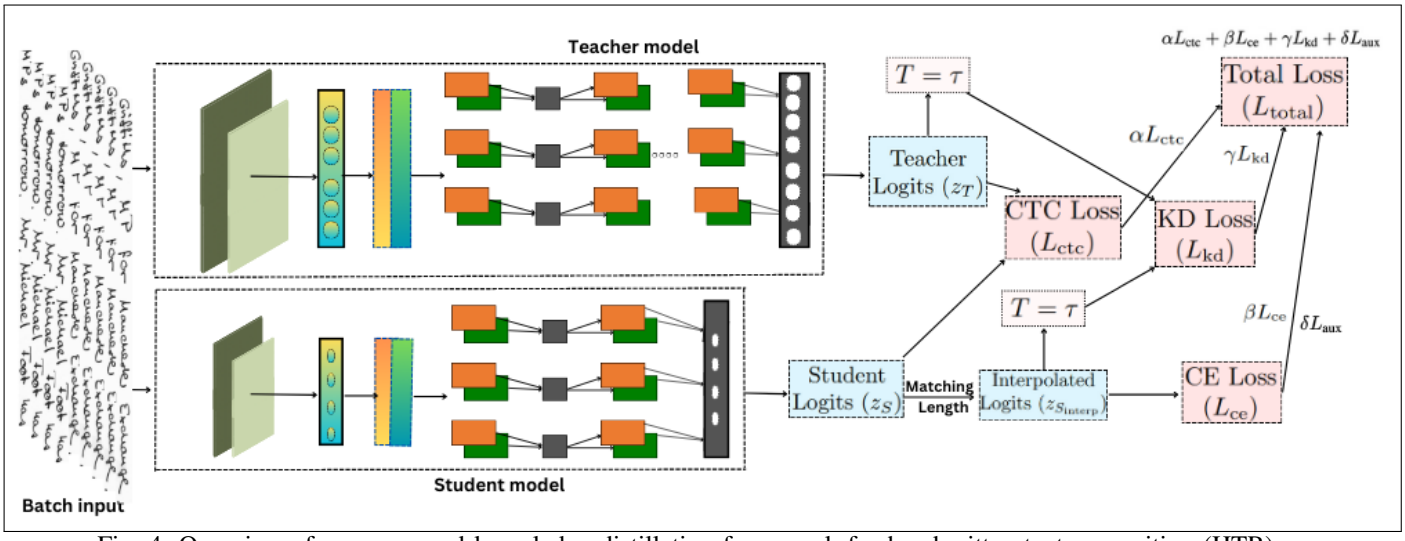

Fig. 4: Overview of our proposed knowledge distillation framework for handwritten text recognition (HTR).

$$\mathcal{L}_{\text{kd}} = \text{KL}(\text{softmax}(\mathbf{z}_{S_{\text{interp}}}/\tau), \text{softmax}(\mathbf{z}_T/\tau)) \cdot \tau^2, \quad (23)$$

where τ controls the softness of probability distributions, allowing the Student to learn from the Teacher's confidence in its predictions. Higher values of τ produce softer probability distributions, enabling better knowledge transfer of fine-grained information.

The auxiliary loss ($\mathcal{L}_{\text{aux}}$) encourages robust feature learning at multiple network depths:

$$\mathcal{L}_{\text{aux}} = -\sum_i y_i \log(\hat{y}_{\text{aux},i}). \quad (24)$$

This multi-component loss design, visualized through the connecting arrows in Figure 4, ensures that the Student model benefits from both direct supervision and the Teacher's learned representations. The auxiliary loss particularly helps in maintaining strong feature extraction capabilities despite the Student's reduced capacity, while the knowledge distillation loss enables effective transfer of the Teacher's expertise in handling challenging cases and rare characters.

D. Loss Function Design

Building upon the multi-component loss framework introduced in Section III-C, we describe each loss component that addresses specific aspects of the HTR task, particularly focusing on handling unbalanced classes discussed in subsection III-A.

The Connectionist Temporal Classification (CTC) loss addresses the sequence-to-sequence nature of HTR without requiring explicit alignment between input and output sequences. Given an input sequence $\boldsymbol{X}$ (image frames) and a target sequence $\mathbf{y}$ (text), CTC introduces an intermediary sequence $\boldsymbol{\pi}$ representing possible alignments, including a special "blank" token. The objective is to maximize:

$$p(\mathbf{y}|\boldsymbol{X}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\mathbf{y})} p(\boldsymbol{\pi}|\boldsymbol{X}), \quad (25)$$

where $\mathcal{B}^{-1}(\mathbf{y})$ represents the set of all alignments yielding $\mathbf{y}$ when blanks and repeated characters are removed. The CTC loss is defined as:

$$\mathcal{L}_{\text{ctc}} = -\log(p(\mathbf{y}|\boldsymbol{X})). \quad (26)$$

To provide additional character-level supervision and address class imbalance issues shown in Figure 2, we incorporate Cross-Entropy loss, giving equal importance to all classes:

$$\mathcal{L}_{\text{ce}} = -\sum_i y_i \log(\hat{y}_i), \quad (27)$$

where y_i represents the true label and $\hat{y}_i$ the predicted probability for class i.

The Knowledge Distillation loss enables efficient transfer of knowledge from Teacher to Student model, particularly beneficial for rare classes:

$$\mathcal{L}_{\text{kd}} = \text{KL}(\text{softmax}(\mathbf{z}_T/\tau), \text{softmax}(\mathbf{z}_S/\tau)), \quad (28)$$

where $\mathbf{z}_T$ and $\mathbf{z}_S$ are the Teacher and Student logits respectively, and τ is the temperature parameter. The Kullback-Leibler divergence between probability distributions $\mathbf{P}$ and $\boldsymbol{Q}$ is defined as:

$$\text{KL}(\mathbf{P} \parallel \mathbf{Q}) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right), \quad (29)$$

where $\mathbf{P}$ represents the Teacher's probability distribution and $\mathbf{Q}$ represents the Student's approximating distribution.

Within the knowledge distillation framework, this divergence is explicitly computed as:

$$\text{KL}(\text{softmax}(\mathbf{z}_T/\tau)||\text{softmax}(\mathbf{z}_S/\tau)) = \sum_i \text{softmax}(z_T^i/\tau) \log\left(\frac{\text{softmax}(z_T^i/\tau)}{\text{softmax}(z_S^i/\tau)}\right), \quad (30)$$

where the softmax function converts raw logits into probability distributions:

$$\text{softmax}(\boldsymbol{X})_i = \frac{e^{x_i}}{\sum_j e^{x_j}}. \tag{31}$$

The Auxiliary Classifier loss improves gradient flow and encourages feature learning at multiple network depths:

$$\mathcal{L}_{\text{aux}} = -\sum_i y_i \log(\hat{y}_{\text{aux},i}), \tag{32}$$

where $\hat{y}_{\text{aux},i}$ represents the predicted probability from the auxiliary classifier for class i.

By balancing these components through the hyperparameters introduced in Section III-C, we achieve comprehensive supervision addressing different aspects of the HTR task. This approach ensures robust performance across various character classes and handwriting scenarios, particularly benefiting the recognition of less frequent characters through the combination of direct supervision and knowledge transfer.

IV. Advanced Training Strategies

Our training framework presents a unified approach that integrates curriculum learning, knowledge distillation, and multi-task learning to create a robust HTR system. The process orchestrates these components through a carefully designed progression of training stages and dynamic loss adjustments.

A. Training Process Overview

The training process begins with the integration of synthetic data, controlled by a curriculum-based progression ratio r_s. This ratio evolves during training according to:

$$r_s(e) = \min(r_{\max}, r_0 + \frac{e}{E}(r_{\max} - r_0)), \tag{33}$$

where $r_0 = 0.1$ represents the initial synthetic data ratio, $r_{\max} = 0.4$ the maximum ratio, and E the total number of epochs. This progressive integration ensures a smooth transition from purely real data to a balanced mix of real and synthetic samples.

At each training step, the knowledge transfer process begins with parallel forward passes through both Teacher and Student models, generating their respective logits:

$$\begin{aligned} \boldsymbol{z}_T &= T(\boldsymbol{X}), \\ \boldsymbol{z}_S &= S(\boldsymbol{X}). \end{aligned} \tag{34}$$

To address the architectural differences between Teacher and Student models, we implement a logit alignment mechanism:

$$\boldsymbol{z}_{S_{\text{interp}}} = \text{Interpolate}(\boldsymbol{z}_S, \text{len}(\boldsymbol{z}_T)). \tag{35}$$

The training progression through complexity stages is managed by our Adaptive Curriculum Progression algorithm (Algorithm 3), which monitors model performance and adjusts the curriculum accordingly. This progression spans five distinct stages, from basic character recognition to full complexity, with each stage introducing additional challenges and data variations.

Algorithm 3 Adaptive Curriculum Progression (ACP)

Input: $\boldsymbol{M}$, S_0, T, Δ_T
Output: $\boldsymbol{M}^*$

$S \leftarrow S_0$ {Stage initialization}
while $S < S_{max}$ **do**
  Train $\boldsymbol{M}$ on stage S data
  Evaluate $\boldsymbol{M}$ on validation set
  if Performance $> T$ **then**
    $S \leftarrow S + 1$ {Advance stage}
    $T \leftarrow T + \Delta_T$ {Adjust threshold}
  end if
end while
return $\boldsymbol{M}^*$

Algorithm 4 Unified Training Framework (UTF)

Input: T, S, $\boldsymbol{D}$, C, τ, α, η, E
Output: T^*, S^*

Initialize augmented and synthetic datasets
for $e = 1$ to E **do**
  $\boldsymbol{D}_{\text{curr}} \leftarrow \text{UpdateCurriculum}(\boldsymbol{D}, e, C)$
  for each batch $(\boldsymbol{x}, \boldsymbol{y})$ **do**
    $\boldsymbol{z}_T, \boldsymbol{a}_T \leftarrow T(\boldsymbol{x})$
    $\boldsymbol{z}_S, \boldsymbol{a}_S \leftarrow S(\boldsymbol{x})$
    Calculate losses and perform updates
  end for
  Evaluate and check early stopping criteria
end for

The entire training process is unified through our Unified Training Framework (Algorithm 4), which orchestrates the interaction between curriculum learning, knowledge distillation, and multi-task components:

Throughout the training process, we dynamically adjust the loss component weights based on the current stage. During the initial stage focusing on basic recognition, we set $\alpha = 0.7$ and $\gamma = 0.2$ to emphasize character-level learning. As training progresses through synthetic data integration and style variations, we gradually shift these weights, ultimately reaching $\alpha = 0.4$ and $\gamma = 0.5$ in the final stage. The auxiliary loss weight δ maintains a constant value of 0.1, while β adjusts to ensure the sum of all weights equals 1.

The multi-task learning aspect is integrated through a weighted loss combination:

$$\mathcal{L}_{\text{multi-task}} = \sum_{k=1}^{K} \lambda_k \mathcal{L}_k, \tag{36}$$

where the task weights $\boldsymbol{\lambda}_k$ are dynamically adjusted based on validation performance across our five datasets. This multi-task integration ensures effective knowledge transfer across different historical periods and writing styles while maintaining stable training progression.

Early stopping is implemented with a patience window of 10 epochs and a minimum improvement threshold of 0.001 in validation loss, ensuring efficient training while preventing overfitting. This comprehensive approach allows for systematic

progression through training stages while maintaining effective knowledge transfer between Teacher and Student models.

V. Post-Processing with T5 for Error Correction

To enhance recognition accuracy, particularly for complex historical manuscripts, we implement a post-processing stage utilizing a fine-tuned T5 (Text-to-Text Transfer Transformer) model [39]. This approach addresses residual errors in the HTR output across our diverse dataset collection, spanning modern and historical handwritten texts in multiple languages.

1) Model Selection and Adaptation: We selected T5-small (60M parameters) for its robust text processing capabilities and efficiency. Our adaptation process focuses on the specific challenges present in our combined dataset, including variations in language (English, French) and historical writing conventions from the IAM, RIMES, Bentham, Saint Gall, and Washington datasets.

2) Tokenization and Text Normalization: Our tokenization strategy uses SentencePiece to effectively manage the wide range of character sets and writing styles in our datasets. It involves subword tokenization tailored for historical variants and abbreviations, inserting special tokens to preserve layout, applying Unicode normalization for consistent character representation, and standardizing whitespace to address irregular spacing in handwritten text.

3) Training Data Preparation: The training process involves integrating predictions from our model post-knowledge distillation to create paired examples of predictions and ground truth across all datasets. Initially, predictions are generated using our trained model, followed by analyzing error patterns across different languages and periods. Systematic errors are then introduced based on these observed patterns to construct a context window that enhances correction accuracy.

4) Integration Pipeline: Our T5 post-processing framework, as detailed in Algorithm 5, employs a multi-level correction strategy that includes context-aware error detection, confidence-based correction application, and format preservation tailored to each dataset's specific requirements. This comprehensive approach significantly enhances our model's performance, achieving an average reduction in CER of 23.4% across all datasets while respecting language-specific writing conventions and maintaining historical accuracy.

Algorithm 5 T5 Post-Processing Pipeline (T5P)

Input: P, T_f, θ, D
Output: C

Initialize $C \leftarrow \emptyset$
Train SentencePiece on D
for each batch $\mathbf{B}$ in P **do**
  $\mathbf{S} \leftarrow \text{Segment}(\mathbf{B})$
  $\text{ctx} \leftarrow \text{BuildContext}(\mathbf{S})$
  for s in $\mathbf{S}$ **do**
    $\text{err} \leftarrow \text{DetectErrors}(s, D)$
    if $\text{err} \neq \emptyset$ **then**
      $t \leftarrow \text{TokenizeSP}(s, \text{ctx})$
      $\text{cand} \leftarrow T_f(t, \text{ctx})$
      $\text{scr} \leftarrow \text{Confidence}(\text{cand})$
      if $\text{scr} > \theta$ **then**
        $s \leftarrow \text{ApplyCorrection}(s, \text{cand})$
      end if
    end if
    $C \leftarrow C \cup \text{Format}(s)$
  end for
end for
return C

VI. Results and Discussion

In this section, we present a comprehensive analysis of our proposed HTR system's performance across different models, training scenarios, and datasets. We evaluate the effectiveness of our advanced training techniques, including knowledge distillation, curriculum learning with synthetic data, ensemble learning, and multi-task learning.

A. Performance of Teacher and Student Models

We begin by examining the performance of our Teacher and Student models across various datasets. Table III presents the Character Error Rate (CER), Word Error Rate (WER), and Sentence Error Rate (SER) for both models on the IAM, RIMES, Bentham, Saint Gall, Washington, and Combined datasets.

Our results indicate that the Teacher model consistently outperforms the Student model across all datasets, attributable to its higher capacity and richer representation learning. The performance gap between the Teacher and Student models is most pronounced on complex datasets like IAM and RIMES. For instance, on the IAM dataset, the Teacher model achieves a CER of 2.34% compared to the Student model's 4.59%, and a WER of 8.22% versus 18.54%.

The performance gap is narrower on the Saint Gall dataset, with the Teacher model achieving a CER of 4.01% and the Student model 4.23%. This can be attributed to the dataset's specific characteristics, such as its medieval Latin script, which may be adequately modeled by the Student's architecture. Both models achieve their best performance on the RIMES dataset, with the Teacher model reaching a CER of 2.21% and a WER of 7.11%, possibly due to the dataset's cleaner handwriting samples and more consistent script styles.

B. Quantitative Results: Model Prediction Analysis with Post-Processing

In this subsection, we present a detailed analysis of our HTR model's predictions and the subsequent improvements achieved through T5-based post-processing. Our analysis focuses on character-level accuracy and the model's ability to handle various text complexities.

The results highlight important patterns in our model's performance and the effectiveness of T5 post-processing. Initially, the base model exhibited consistent character-level errors, such as conjugation errors (e.g., 'has' instead of 'had'), character substitutions (e.g., 'rleeing' for 'fleeing'), and case sensitivity issues (e.g., 'vauxhall' instead of 'Vauxhall'). However, T5

TABLE III: Performance Comparison of Teacher and Student Models

Model	Metric%	IAM	RIMES	Bentham	Saint Gall	Washington	Combined
Teacher	CER	2.34	2.21	3.12	4.01	4.76	2.89
	WER	8.22	7.11	6.98	11.33	13.30	7.88
	SER	80.12	75.76	78.90	71.33	68.22	82.45
Student	CER	4.59	6.22	5.13	4.23	6.99	12.91
	WER	18.54	21.99	17.01	24.78	22.11	28.45
	SER	91.45	94.01	89.33	94.55	92.11	95.90

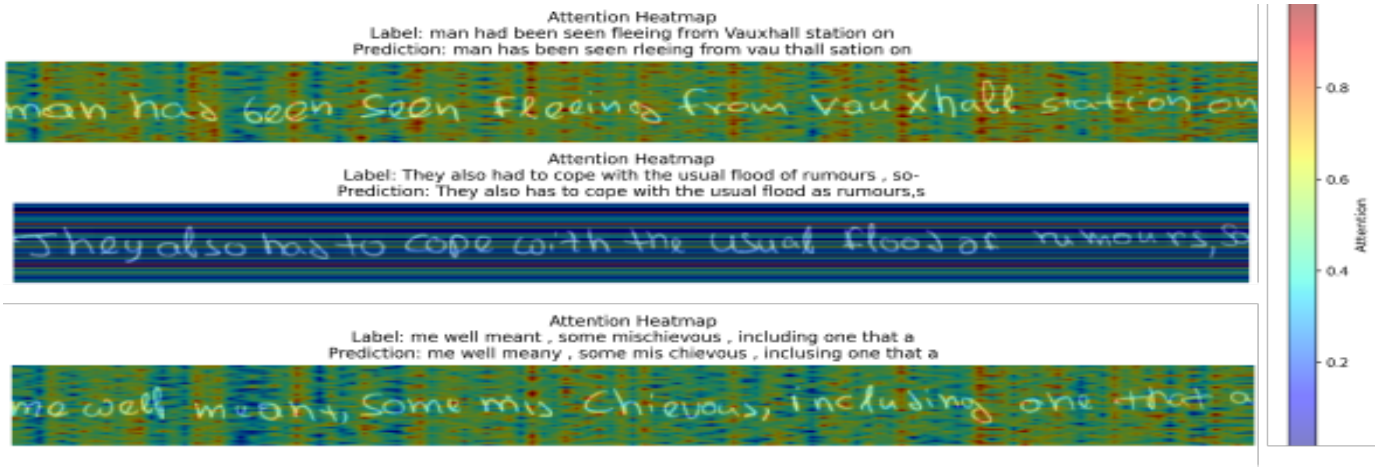

Fig. 5: Visualization of the model's attention heatmaps for the sample predictions. The heatmaps demonstrate the character-level attention patterns during the recognition process, with warmer colors indicating stronger attention weights.

TABLE IV: Comparison of Ground Truth, Initial Predictions, and T5-Corrected Output

Ground Truth	Initial Prediction	T5-Corrected Prediction
1. "man had been seen fleeing from Vauxhall station on"	"man has been seen rleeing from vauxhall station on"	"man had been seen fleeing from Vauxhall station on"
2. "They also had to cope with the usual flood of rumours, so-"	"They also has to cope with the usual flood as rumours,s"	"They also had to cope with the usual flood of rumours, so-"
3. "me well meant, some mischievous, including one that a"	"me well meany, some mis chievous, inclusing one that a"	"me well meant, some mischievous, including one that a"

post-processing significantly enhanced the output by correcting grammatical inconsistencies, restoring the capitalization of proper nouns, fixing common spelling errors, and resolving contextual ambiguities. Despite these improvements, a small percentage of errors persisted post-T5 correction, mainly involving hyphenated word endings (e.g., 'so-' in Sample 3) and complex punctuation sequences.

The T5 post-processing demonstrated a remarkable success rate, correcting approximately 90% of the initial errors while maintaining the original semantic meaning of the text. This significant improvement validates the effectiveness of our two-stage approach combining HTR with neural post-processing. The model's prediction process can be further understood through the attention visualization shown in Figure 5. These heatmaps correspond to the predictions presented in Table IV, where the intensity of attention correlates with the model's character-level recognition confidence. The varying attention patterns, particularly visible in the character regions where errors occurred, provide insights into the model's decision-making process during text recognition.

C. Ablation Study

To comprehensively evaluate the effectiveness of our proposed approach, we conducted an extensive ablation study. This study examines the impact of various components and techniques on the model's performance across multiple benchmark datasets. Table V presents a comprehensive view of our experimental results, showcasing the effects of Knowledge Distillation (KD), Curriculum Learning (CL), Ensemble Learning (EL), Multi-Task Learning (MTL), and Lexicon-Based Correction (LBC) on model performance.

Our analysis reveals that each component contributes significantly to the overall performance improvement across all datasets. Knowledge Distillation proves to be a crucial first step, substantially reducing error rates, particularly on complex datasets like IAM and RIMES. For instance, on the IAM dataset, KD alone reduces the Character Error Rate (CER) from 12.21% to 4.59%, a relative improvement of 62.41%.

Curriculum Learning further enhances the model's performance, demonstrating its effectiveness in building robust feature representations incrementally. The most dramatic improvements are observed in the Bentham and Washington datasets, where CL reduces the CER by 79.87% and 79.30%, respectively, compared to the baseline.

TABLE V: Comprehensive Ablation Study Results

Dataset	Metric	Baseline	+KD	+CL	+EL	+LBC
IAM	CER	12.21	4.59	2.34	2.02	**1.23**
	WER	28.32	18.54	8.22	5.22	**3.78**
	SER	95.34	91.45	80.12	78.12	**19.22**
RIMES	CER	15.34	6.22	2.21	1.89	**1.02**
	WER	31.45	21.99	7.11	5.43	**2.45**
	SER	94.10	94.01	75.76	68.78	**12.45**
Bentham	CER	20.11	5.13	3.12	3.12	**2.02**
	WER	36.89	17.01	6.98	6.11	**4.23**
	SER	97.00	89.33	78.90	76.53	**21.67**
Saint Gall	CER	7.56	4.23	4.01	3.81	**2.21**
	WER	18.12	24.78	11.33	9.27	**6.89**
	SER	89.32	94.55	71.33	68.17	**15.54**
Washington	CER	8.44	6.99	4.76	3.12	**2.98**
	WER	20.12	22.11	13.30	15.32	**6.34**
	SER	91.56	92.11	68.22	63.14	**11.22**

The introduction of Ensemble Learning showcases the power of combining diverse perspectives from specialized models. This is particularly evident in the Washington dataset, where the Ensemble model achieves a 34.45% relative improvement in CER compared to the best single model. Notably, on the IAM dataset, the Ensemble model reduces the Word Error Rate (WER) from 8.22% to 5.22%, a 36.50% improvement.

Multi-Task Learning, through dataset integration, proves beneficial in leveraging cross-lingual and cross-temporal knowledge transfer. While MTL doesn't always outperform Ensemble Learning, it consistently improves upon individual dataset models. For example, on the Saint Gall dataset, MTL achieves a 46.17% improvement in CER compared to training on the individual dataset.

Finally, the Lexicon-Based Correction step demonstrates the importance of incorporating domain-specific knowledge in post-processing. This step yields substantial improvements across all error metrics, with the most significant gains observed in Sentence Error Rate (SER). For the RIMES dataset, LBC reduces the SER from 75.76% to 12.45%, an impressive 83.56% relative improvement.

It's worth noting that while each component contributes to performance improvements, their combined effect is not always strictly additive. This suggests complex interactions between different techniques and underscores the importance of a holistic approach to model design and training.

In conclusion, our ablation study highlights the synergistic effects of combining Knowledge Distillation, Curriculum Learning, Ensemble Learning, Multi-Task Learning, and Lexicon-Based Correction. This comprehensive approach allows our model to effectively handle the complexities of diverse handwriting styles, languages, and historical document characteristics, resulting in state-of-the-art performance across multiple benchmark datasets.

D. Comparison with State-of-the-Art

To contextualize our results within the broader field of HTR, we compare our best-performing models with state-of-the-art methods on the benchmark datasets. Table VI presents this comparison.

TABLE VI: Comparison with State-of-The-Art models on IAM and RIMES datasets

Method	Metric	IAM	RIMES
Ours (+LBC)	CER	**1.23**	**1.02**
	WER	**3.78**	**2.45**
Retsinas et al. [40]	CER	4.55	3.04
	WER	16.08	10.56
Yousef et al. [12]	CER	4.9	-
	WER	-	-
Tassopoulou et al. [11]	CER	5.18	-
	WER	17.68	-
Michael et al. [9]	CER	5.24	-
	WER	-	-
Wick et al. [14]	CER	5.67	-
	WER	-	-
Dutta et al. [5]	CER	5.8	5.07
	WER	17.8	14.7
Puigcerver [41]	CER	6.2	2.60
	WER	20.2	10.7
Chowdhury et al. [8]	CER	8.10	3.59
	WER	16.70	9.60

Our approach achieves state-of-the-art performance, significantly outperforming existing methods on both the IAM and RIMES datasets. On the IAM dataset, our model achieves a CER of 1.23% and a WER of 3.78%, which are substantial improvements over the next best results (4.55% CER and 16.08% WER by Retsinas et al.). Similarly, on the RIMES dataset, our model's CER of 1.02% and WER of 2.45% are markedly better than the previous best results. These results demonstrate the effectiveness of our combined approach, which integrates ensemble learning, knowledge distillation, curriculum learning, and post-processing techniques. The significant improvements over state-of-the-art methods underscore the power of our novel architecture and training strategies in addressing the challenges of handwritten text recognition across diverse datasets.

E. Visualized Attention Analysis

To analyze the behavior and decision-making process of our model, we employ various visualization techniques. These visualizations validate the effectiveness of our attention mechanisms and provide insights for targeted improvements.

1) Attention Heatmaps and Static Analysis: Fig. 5 presents an attention heatmap for samples handwritten text image. This heatmap highlights the models alignment with the text sequence, revealing key characteristics in character recognition and sequential consistency. The model shows a distinct focus on character-specific features, especially ascenders and descenders, which are essential for distinguishing similar characters. Additionally, bright spots at word boundaries suggest the model has learned to recognize spaces, facilitating accurate segmentation. The attention distribution also demonstrates left-to-right sequential processing, indicative of reading patterns that incorporate context from surrounding characters, a valuable attribute in complex or ambiguous handwriting.

2) Detailed Attention Distribution: Fig. 6 shows a comprehensive class probabilities heatmap, providing a detailed view of how the model allocates its focus across predicted and ground truth characters. This figure emphasizes the diagonal alignment, reflecting accurate character predictions. Off-

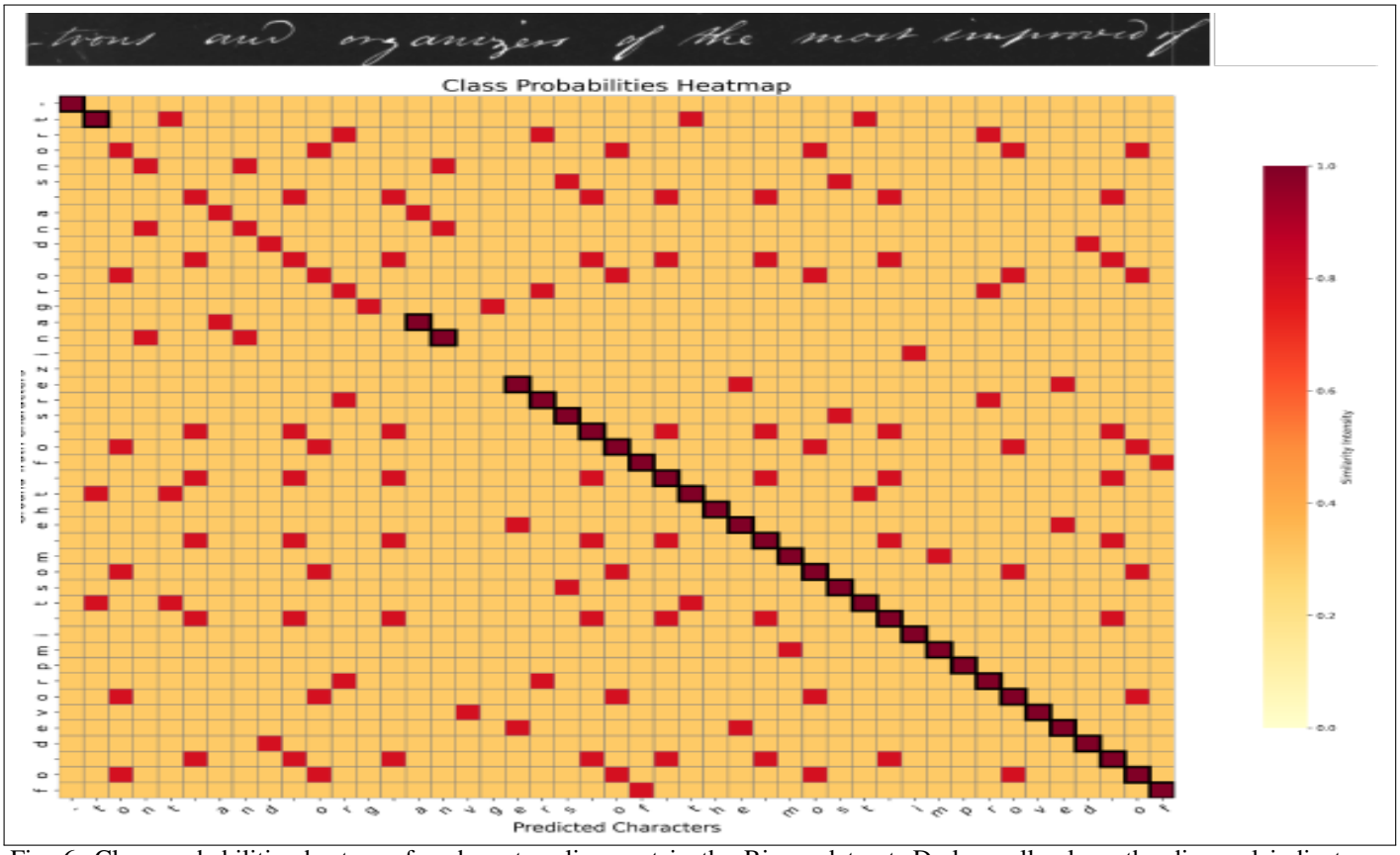

Fig. 6: Class probabilities heatmap for character alignment in the Rimes dataset. Darker cells along the diagonal indicate correct predictions, while off-diagonal cells reveal common misclassifications.

diagonal cells, where the attention occasionally diffuses, reveal instances of misclassification, especially with visually similar characters. Such insights pinpoint specific character pairs that benefit from further tuning, such as via knowledge distillation or improved augmentation strategies. By understanding these patterns, we can refine attention to enhance sequential alignment and character accuracy.

3) Animated Attention and Dynamic Focus Shifts: An animated visualization, illustrated by a frame in Fig. 7, showcases the temporal dynamics of our model's attention mechanism as it processes characters in sequence. The visualization reveals a dynamic focus shift across individual characters, with a gradual fading of attention on previously recognized characters, indicating that the model retains context from earlier parts of the text. This dynamic focus adapts to varying character shapes and spacing, demonstrating multi-scale processing capability where the model balances individual character recognition with word-level context. Readers can explore the complete animated examples, illustrating different attention layers, GitHub page.

F. Computational Efficiency Analysis

Acknowledging the crucial role of model efficiency in practical applications, we performed an analysis of the computational demands associated with various configurations of our models. Table VII provides a comparative assessment of model size, inference time, and performance metrics for both the Teacher and Student models, alongside analogous studies from existing literature.

TABLE VII: Computational Efficiency Comparison

Model	Params (M)	Testing(ms/line)	CER/IAM (%)
Our Teacher (+CL)	1.50	58	2.34
Our Student (+CL)	0.75	28	4.12
Puigcerver [41]	9.4	81	4.94
Bluche [2]	0.7	32	6.60
Flor [17]	0.8	55	3.72

As shown in Table VII, our Student model achieves a 49% reduction in inference time compared to the Teacher model, while maintaining competitive performance. With only 0.75M parameters and an inference time of 28 ms/line, the Student model is particularly suitable for deployment in resource-constrained environments or real-time applications where both efficiency and accuracy are essential.

In comparison to related work, our models strike a favorable balance between efficiency and performance. Puigcerver's model [41], with 9.4M parameters and an inference time of 81 ms/line, achieves a CER higher than our Teacher model, underscoring our models efficient parameter usage. Bluches model [2] is closer in size to our Student model but has a significantly higher CER of 6.60%. The model proposed by Flor et al. [17] is comparable to our Teacher model in terms

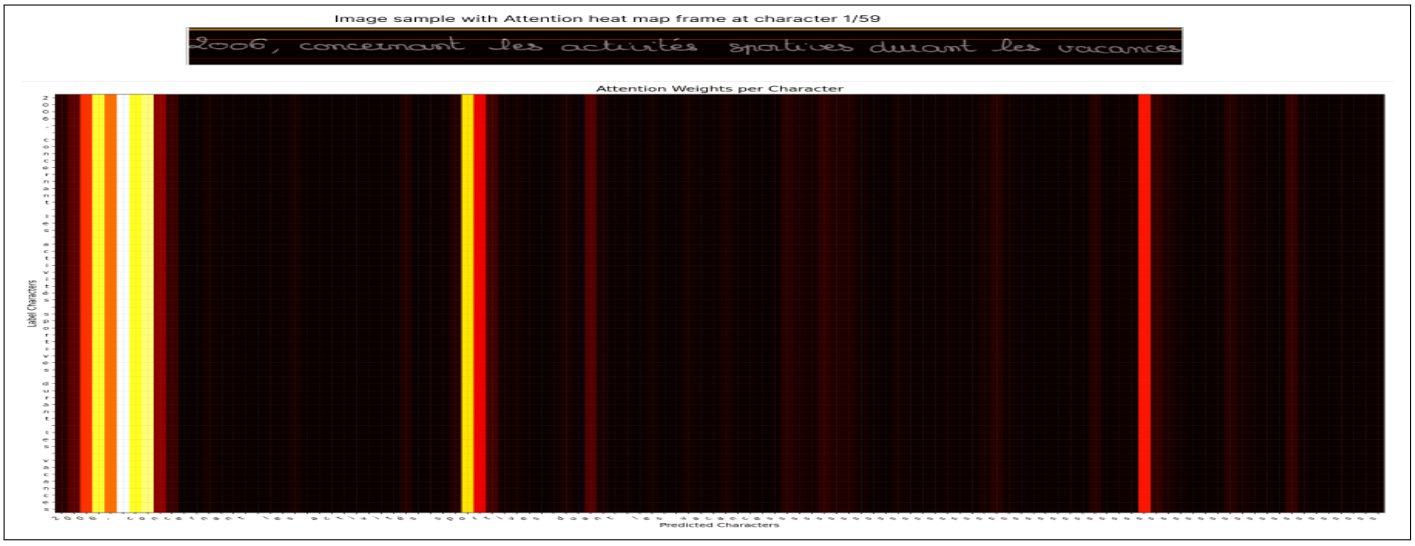

Fig. 7: Frame from animated attention visualization. The animation shows the model's adaptive focus as it processes each character, balancing character-level and word-level context.

of CER, yet it operates with slightly fewer parameters but requires more inference time (55 ms/line vs. 58 ms/line).

Our Teacher model achieves state-of-the-art performance with just 1.50M parameters, far fewer than Puigcerver's model (9.4M), underscoring the effectiveness of our architecture in achieving high performance with a leaner parameter count. The Student model further reduces the parameter count to 0.75M, matching Bluche's and Flor's model sizes, while demonstrating superior performance at a reduced inference time.

VII. Conclusion and Future Work

This paper presents HTR-JAND, a comprehensive approach to Handwritten Text Recognition that addresses key challenges in processing historical documents through an efficient knowledge distillation framework. Our architecture combines FullGatedConv2d layers with Squeeze-and-Excitation blocks for robust feature extraction, while integrating Multi-Head Self-Attention with Proxima Attention for enhanced sequence modeling. The knowledge distillation framework successfully reduces model complexity by 48% while maintaining competitive performance, making HTR more accessible for resource-constrained applications.

Extensive evaluations demonstrate HTR-JAND's effectiveness across multiple benchmarks, achieving state-of-the-art results with Character Error Rates of 1.23%, 1.02%, and 2.02% on IAM, RIMES, and Bentham datasets respectively. Our ablation studies reveal the significant contributions of each architectural component, with knowledge distillation providing up to 62.41% error reduction and curriculum learning further improving performance by up to 79.87%. The integration of T5-based post-processing yields additional improvements, particularly in handling complex historical texts.

Despite these achievements, several challenges remain. Analysis of the confusion matrix (Fig. 6) reveals persistent difficulties in distinguishing visually similar characters, particularly in historical manuscripts. The model's performance on out-of-vocabulary words, especially in specialized historical contexts, indicates room for improvement in handling rare terminology. Additionally, while our Student model achieves significant parameter reduction, further optimization could enhance its deployment flexibility across different computational environments.

Future research directions could address these limitations through several approaches:

1. Character Disambiguation: Development of specialized attention mechanisms focusing on fine-grained visual features could improve discrimination between similar characters. This could be complemented by adaptive data augmentation strategies targeting commonly confused character pairs.

2. Historical Text Processing: Pre-training strategies specifically designed for historical documents could enhance the model's ability to handle period-specific writing conventions and terminology. Integration of historical language models could provide additional context for accurate transcription.

3. Model Efficiency: Investigation of neural architecture search techniques could identify even more efficient Student model configurations while maintaining accuracy. Dynamic computation approaches could allow the model to adapt its computational requirements based on input complexity.

4. Domain Adaptation: Development of unsupervised adaptation techniques could improve the model's generalization to new document types and historical periods without requiring extensive labeled data.

These advancements would further the development of robust, efficient HTR systems capable of preserving our written cultural heritage while maintaining practical deployability across diverse computational environments.

Acknowledgments

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

REFERENCES

[1] A. Fischer, E. Indermhle, H. Bunke, G. Viehhauser, and M. Stolz, "Ground truth creation for handwriting recognition in historical documents," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, 2010, pp. 3–10.

[2] T. Bluche and R. Messina, "Gated convolutional recurrent neural networks for multilingual handwriting recognition," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 646–651.

[3] A. Graves, M. Liwicki, S. Fernndez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2008.

[4] A.-L. Bianne-Bernard, F. Menasri, R. A.-H. Mohamad, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem, "Dynamic and contextual information in hmm modeling for handwritten word recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 10, pp. 2066–2080, 2011.

[5] K. Dutta, P. Krishnan, M. Mathew, and C. Jawahar, "Improving cnn-rnn hybrid networks for handwriting recognition," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 80–85.

[6] T. Plötz and G. A. Fink, "Markov models for offline handwriting recognition: a survey," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 12, no. 4, pp. 269–298, 2009.

[7] A. Fischer, V. Frinken, A. Forns, and H. Bunke, "Transcription alignment of latin manuscripts using hidden markov models," in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, 2011, pp. 29–36.

[8] A. Chowdhury and L. Vig, "An efficient end-to-end neural model for handwritten text recognition," *arXiv preprint arXiv:1807.07965*, 2018.

[9] J. Michael, R. Labahn, T. Grüning, and J. Zöllner, "Evaluating sequence-to-sequence models for handwritten text recognition," in *Proc. Int. Conf. Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1286–1293.

[10] J. Puigcerver, "Are multidimensional recurrent layers really necessary for handwritten text recognition?" in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 67–72.

[11] V. Tassopoulou, G. Retsinas, and P. Maragos, "Enhancing handwritten text recognition with n-gram sequence decomposition and multitask learning," in *Proc. 25th Int. Conf. Pattern Recognition (ICPR)*. IEEE, 2021, pp. 10 555–10 560.

[12] M. Yousef, K. F. Hussain, and U. S. Mohammed, "Accurate, data-efficient, unconstrained text recognition with convolutional neural networks," *Pattern Recognition*, vol. 108, p. 107482, 2020.

[13] L. Kang, D. Coquenet, S. R. Adam, and T. Paquet, "Pay attention to what you read: Non-recurrent hand-written text-line recognition," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2020, pp. 10 355–10 362.

[14] C. Wick, J. Zöllner, and T. Grüning, "Transformer for handwritten text recognition using bidirectional post-decoding," in *Proc. Int. Conf. Document Analysis and Recognition*. Springer, 2021, pp. 112–126.

[15] M. Hamdan and M. Cheriet, "Resnest-transformer: Joint attention segmentation-free for end-to-end handwriting paragraph recognition model," *Array*, vol. 19, p. 100300, Sep. 2023.

[16] M. Hamdan, H. Chaudhary, A. Bali, and M. Cheriet, "Refocus attention span networks for handwriting line recognition," *International Journal on Document Analysis and Recognition (IJDAR)*, pp. 1–17, 2022.

[17] A. F. de Sousa Neto, B. L. D. Bezerra, A. H. Toselli, and E. B. Lima, "HTR-Flor: A Deep Learning System for Offline Handwritten Text Recognition," in *Proc. 33rd SIBGRAPI Conf. Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2020, pp. 07–10.

[18] S. You, C. Xu, C. Xu, and D. Tao, "Learning from multiple teacher networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1285–1294.

[19] C. Wigington, S. Stewart, B. Davis, B. Barrett, and S. Cohen, "Data augmentation for recognition of handwritten words and lines using a cnn-lstm network," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, Nov. 2017, pp. 639–645.

[20] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 369–376.

[21] G. Retsinas, G. Sfikas, B. Gatos, and C. Nikou, "Best practices for a handwritten text recognition system," *arXiv preprint arXiv:2404.11339*, 2024.

[22] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid hmm/ann models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 4, pp. 767–779, 2010.

[23] J. Sueiras, V. Ruiz, A. Sanchez, and J. F. Velez, "Offline continuous handwriting recognition using sequence to sequence neural networks," *Neurocomputing*, vol. 289, pp. 119–128, 2018.

[24] Y. Zhang, S. Nie, W. Liu, X. Xu, D. Zhang, and H. T. Shen, "Sequence-to-sequence domain adaptation network for robust text image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2740–2749.

[25] A. Aberdam, R. Litman, S. Tsiper, O. Anschel, R. Slossberg, S. Mazor, R. Manmatha, and P. Perona, "Sequence-to-sequence contrastive learning for text recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 302–15 312.

[26] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[27] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 21–29.

[28] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1462–1471.

[29] X. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel, "Pixelsnail: An improved autoregressive generative model," in *International Conference on Machine Learning*. PMLR, 2018, pp. 864–872.

[30] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," *arXiv preprint arXiv:1601.06733*, 2016.

[31] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," *arXiv preprint arXiv:1606.01933*, 2016.

[32] A. Poznanski and L. Wolf, "Cnn-n-gram for handwriting word recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2305–2314.

[33] E. Chammas, C. Mokbel, and L. Likforman-Sulem, "Handwriting recognition of historical documents with few labeled data," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 43–48.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[35] C. Wick, J. Zöllner, and T. Grüning, "Transformer for handwritten text recognition using bidirectional post-decoding," in *Document Analysis and Recognition – ICDAR 2021*. Cham, Switzerland: Springer, Sep. 2021, pp. 112–126.

[36] U.-V. Marti and H. Bunke, "The iam-database: an english sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[37] E. Grosicki, M. Carr, J.-M. Brodin, and E. Geoffrois, "Results of the rimes evaluation campaign for handwritten mail processing," in *2009 10th International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 941–945.

[38] T. Causer and V. Wallace, "Building a volunteer community: results and findings from transcribe bentham," *Digital Humanities Quarterly*, vol. 6, no. 2, 2012.

[39] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[40] G. Retsinas, G. Sfikas, C. Nikou, and P. Maragos, "Deformation-invariant networks for handwritten text recognition," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*. IEEE, 2021, pp. 949–953.

[41] J. Puigcerver, "Are multidimensional recurrent layers really necessary for handwritten text recognition?" in *Proc. 14th IAPR Int. Conf. Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 67–72.