SemiHastakshar: Generalizable Indic Handwritten OCR through Semi-Supervised Learning

Lalitha Evani
CVIT, IIIT-Hyderabad
India
lalitha.e@research.iiit.ac.in

Ajoy Mondal
CVIT, IIIT-Hyderabad
India
ajoy.mondal@iiit.ac.in

C. V. Jawahar
CVIT, IIIT-Hyderabad
India
jawahar@iiit.ac.in

Abstract

Handwritten Text Recognition (HTR) remains a challenging task, especially in Indic languages, due to varied handwriting styles, complex character structures, and the scarcity of annotated data. While deep learning models have shown promising results on curated benchmarks, their performance often degrades in real-world scenarios with high variability in writing. To address this, we propose a semi-supervised approach, **SemiHastakshar**, that leverages large-scale unlabeled word-level images collected from diverse sources across the Internet. We employ a high-confidence pseudo-labeling strategy to train on unlabeled samples iteratively. It allows the model to learn from a wider distribution of handwriting styles and improve generalization across writers. Our method demonstrates that combining a small amount of labeled data with a large unlabeled corpus leads to more robust HTR models for Indic scripts, advancing scalable recognition in low-resource settings. The code, model, and data will be publicly available for research.

CCS Concepts

• **Applied computing** → **Document analysis**; **Optical character recognition**.

Keywords

Handwritten, Text Recognition, Indian Languages, Unlabeled Data, Semi-supervised Learning, Large Corpus.

ACM Reference Format:
Lalitha Evani, Ajoy Mondal, and C. V. Jawahar. 2025. SemiHastakshar: Generalizable Indic Handwritten OCR through Semi-Supervised Learning. In *Proceedings of 16th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'25).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3774521.3774605

1 Introduction

The journey of understanding language begins with the humble symbol. It is a mark that carries meaning, emotion, and thought. When these symbols are arranged meaningfully, they form the basis of written communication, enabling comprehension across generations, cultures, and contexts. The ability to recognize and interpret these symbols from visual input lies at the heart of Optical Character Recognition (OCR). This field seeks to convert images containing text into machine-readable formats.

A key subset of OCR is Handwritten Text Recognition (HTR), which deals with deciphering free-form handwriting, a profoundly personal, stylistically rich, and often beautifully idiosyncratic medium. Unlike printed text, handwriting is shaped by individual quirks, cultural influences, and the writer's motor behavior, making it uniquely expressive and inherently challenging to model [31]. In HTR, every writer introduces a new style, and every character instance might vary in shape, curvature, and structure, even within the same document.

These challenges become significantly more pronounced when considering Indic languages. With over 20 official scripts and hundreds of spoken languages, India represents one of the most linguistically diverse regions in the world [19]. Each script, Devanagari, Bangla, Tamil, Telugu, or others, has a rich character set, intricate ligatures, and complex diacritic systems that pose non-trivial difficulties for recognition systems. This work systematically focuses on multiple Indic scripts to explore these challenges.

Indic handwritten text recognition is a complex problem due to the diversity of handwriting across individuals and the inherent structural complexity of Indic scripts. Studies in writer identification [34] and forensic handwriting analysis [21, 38] have demonstrated how handwriting can serve as a biometric marker, an authentication tool, highlighting the degree to which handwriting varies between individuals. This writer-specific variation makes it difficult for HTR models to generalize, especially when trained on limited data collected from a small set of writers. Models trained in such settings are often brittle and fail to perform well when exposed to new, unseen handwriting styles [16].

A key limitation in advancing Indic HTR is the scarcity of annotated data. Curating high-quality labeled datasets across multiple scripts and writers is both labor-intensive and expensive. Moreover, the representation of handwriting styles remains constrained when only a handful of writers contribute to the training set. As a result, models often fail to learn the broad diversity required for real-world robustness. Generalization to new writers remains a significant bottleneck.

To overcome this challenge, we propose a semi-supervised learning approach, **SemiHastakshar**, that leverages the abundance of unlabeled handwritten data available across digital platforms. Annotating every instance of such data is infeasible, and relying exclusively on labeled data limits the model's exposure to stylistic diversity. Instead, we utilize high-confidence pseudo-labeling to iteratively train the model on unlabeled word-level images, allowing it to learn from a broader distribution of handwriting styles without manual supervision. By embracing the diversity encoded

ICVGIP'25, Mandi, India

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/3774521.3774605

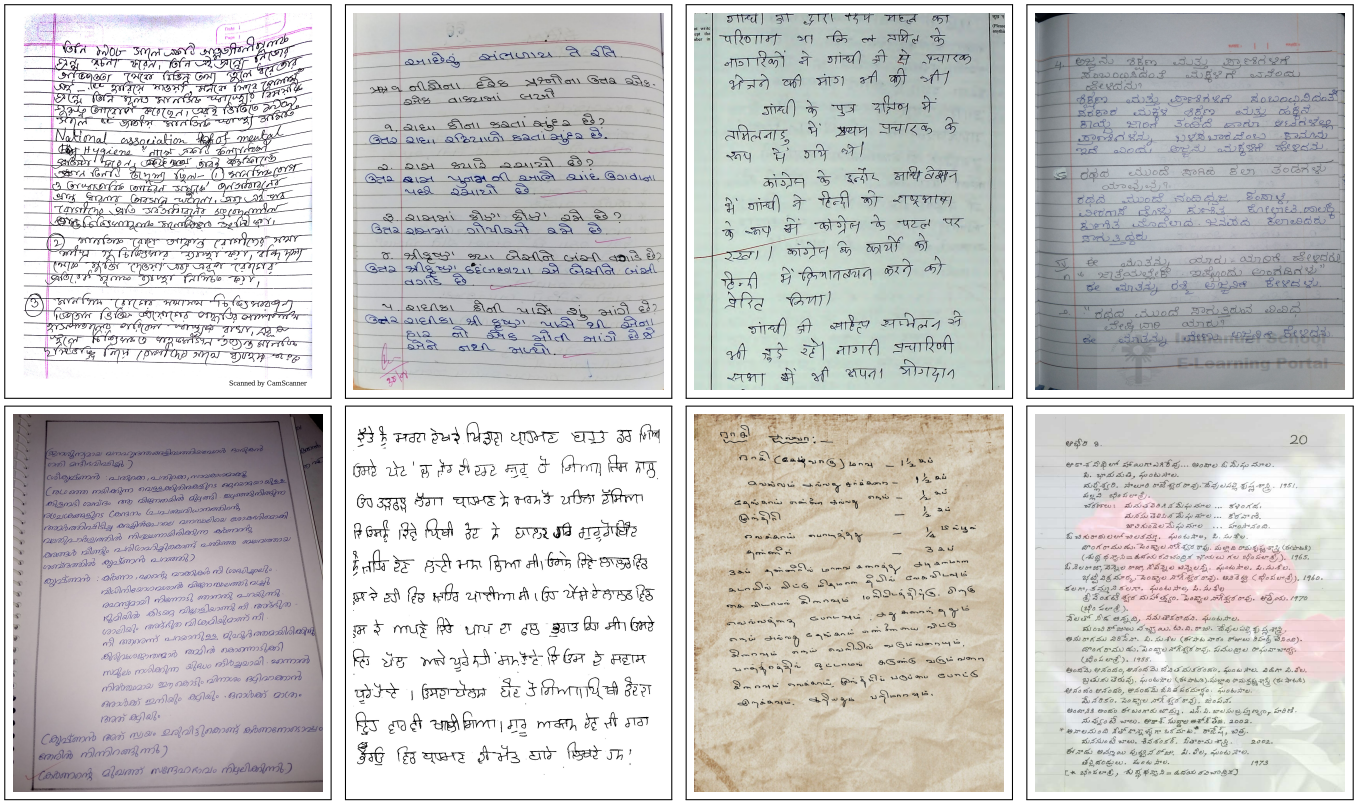

Figure 1: Sample page-level handwritten images from the Indic-HW-Wild dataset, featuring text on both ruled and plain pages. The samples exhibit diverse distortions, including illumination variations, overlapping text, perspective distortions, and noisy backgrounds.

in unlabeled data and focusing on improving cross-writer generalization, our approach paves the way toward scalable and robust HTR systems for Indic scripts. These systems are more adaptable, inclusive, and ready for real-world applications in low-resource scenarios.

Our key contributions are summarized below:

- We present **Indic-HW-Wild**, a large-scale collection of unlabeled word-level handwritten images spanning nine Indic languages — Hindi, Bengali, Telugu, Malayalam, Tamil, Kannada, Odia, Gujarati, and Punjabi — sourced from diverse online repositories.
- We develop a semi-supervised learning framework to train **SemiHastakshar**, a generalized multi-script OCR model that leverages these unlabeled samples to enhance recognition across varied handwriting styles.
- We demonstrate that **SemiHastakshar** achieves significant accuracy gains during dataset-specific fine-tuning, even without target dataset labels, enabling personalized Indic handwriting recognition in low-resource settings.
- Through an extensive ablation study, we quantify the impact of labeled data on model performance, providing insights into data–performance trade-offs.

2 Related Work

Early methods in Indic Handwritten Text Recognition (HTR) predominantly relied on supervised learning, where data underwent pre-processing, segmentation, and manual annotation before being used for model training and evaluation. These supervised approaches can be broadly categorized into three types: segmentation-based, segmentation-free methods, and sequence modeling.

Early approaches were primarily segmentation-based, where handwritten word images were explicitly segmented into individual characters, which were then recognized using classifiers such as Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs) [2, 4, 23]. Roy *et al.* [35] improved this by segmenting into zones and using SVMs and HMMs for zone-specific recognition.

To circumvent the challenges of explicit segmentation, subsequent methods embraced a holistic approach that directly recognized entire word images without requiring character-level boundaries [22, 39, 40]. Shaw *et al.* [40] used sliding windows and chain-code histograms with HMMs, while their later work [39] fused multiple features for improved accuracy. Despite their successes, these holistic methods often rely on fixed-size lexicons, which limit their scalability to larger and more diverse vocabularies.

Deep learning-based architectures have revolutionized Indic HTR by modeling it as a sequence-to-sequence problem. Techniques

based on Bidirectional Long Short-Term Memory with Connectionist Temporal Classification (BLSTM-CTC) and hybrid CNN-RNN frameworks enable recognition of variable-length text without the need for character segmentation [1, 13–15, 20, 37]. Adak *et al.* [1] and Garain *et al.* [15] focused on Bengali. In contrast, others tackled multiple Indic scripts [13, 14, 18]. Recent work has continued in this supervised learning direction, introducing transformer-based models that eliminate the use of CNNs and RNNs by using self-attention mechanisms to model long-range dependencies.

One such transformer-based model inspired by Vision Transformers [12] is PARSeq [7]. Lalitha et al. in [25], use this model to recognize handwritten words in Indian languages using the IIIT-INDIC-HW-WORDS dataset introduced in [18]. While the results in [25] are promising, the dataset has a key limitation: a small number of writers per language. This lack of writer diversity negatively impacts the generalization capability of models, especially given the high variability in handwriting across different Indic scripts. Unlike printed or scene text, handwritten data, in addition to requiring huge amounts of data like other deep learning models, also requires broader stylistic coverage to ensure robustness. Although the collection of large amounts of handwritten data in various Indic languages is a challenge, along with it comes the process of manual annotation, which is time-intensive, depends on the language experts, and is costly. Keeping all these in mind, in this work, we move towards semi-supervised text recognition techniques that utilize the capability of unlabeled data directly along with the high-quality labeled data.

Semi-supervised text recognition utilizes large volumes of unlabeled data alongside limited labeled data to improve model accuracy and generalization. Yang *et al.* [43] provide a detailed survey of SSL methods, broadly categorizing them into deep generative approaches [10, 29, 41], consistency regularization methods [24, 33, 36, 42], and pseudo-labeling techniques [8, 9, 11, 26]. Among these, pseudo-labeling (PL) has emerged as a simple yet effective method. In PL, a model is first trained on the available labeled data, then used to generate predictions for the unlabeled set, which are treated as pseudo-labels. These pseudo-labeled samples are combined with the original labeled set to retrain the model.

Pseudo-labeling has shown promise in domains such as scene text recognition, which was used to incorporate semi-supervised techniques into real-world labeled datasets for the first time [26]. Building on the original PL formulation, Baek *et al.* [5] demonstrated that leveraging a small set of real labeled samples with a strong pretrained model can yield competitive performance and enhanced generalization without the need for extensive annotation. Motivated by these findings, our work applies the pseudo-labeling framework to handwritten text recognition, utilizing collected and pre-processed unlabeled data to achieve robust performance.

3 Indic-HW-Wild Dataset

This work introduces a new dataset, **Indic-HW-Wild**, comprising handwritten word images collected from various internet sources , including publicly available datasets [3, 17, 28, 30, 32]. The primary objective is to curate data across nine Indic scripts captured in unconstrained, real-world conditions, where handwriting quality and image acquisition (e.g., lighting, resolution, scanning quality) vary significantly. It aligns with our broader aim of utilizing unlabeled, in-the-wild data to build robust recognition systems.

Table 1: Language-wise statistics of our newly created unlabeled dataset Indic-HW-Wild for semi-supervised training.

Script	#Pages	#Writers	#Word Instances
Hindi	29,424	90	1,628,801
Bengali	1,709	532	163,492
Telugu	2,609	205	301,170
Gujarati	638	136	59,844
Tamil	1,359	92	146,652
Odia	172	40	9,354
Gurumukhi	132	50	9,362
Malayalam	976	36	57,062
Kannada	1,149	95	73,323

The overall data collection and pre-processing pipeline is illustrated in Fig 2. Initially, we retrieve handwritten content from the web as scanned PDFs or individual page-level images. In cases where PDFs are obtained, they are converted into high-resolution page images. Subsequently, we employ a series of script-agnostic heuristics tailored to each image type to localize and extract handwritten regions of interest.

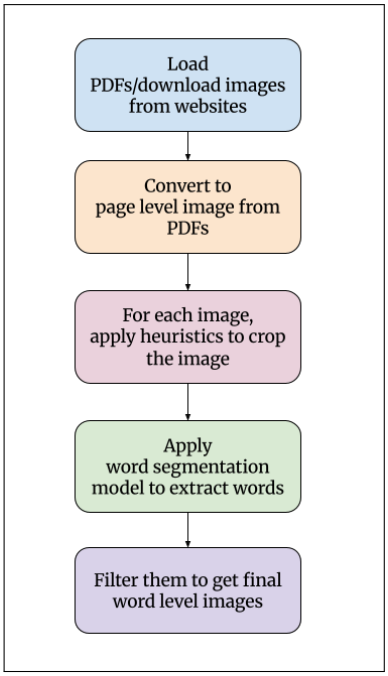

Figure 2: Pipeline for constructing the Indic-HW-Wild dataset, including data collection from diverse online sources, pre-processing to isolate word-level handwritten text, and quality filtering to ensure script and content diversity across nine Indic languages.

We use the CRAFT model [6], which has shown effectiveness in text detection across scripts to perform word-level segmentation. The segmented word images are then subjected to a filtering stage,

where we discard overly small segments or visual outliers to improve the overall quality and consistency of the dataset. After all pre-processing steps, the resulting dataset is summarized in Table 1, showcasing the number of pages, writers, and word instances for each language. Some page and word images from this dataset are shown in Figs. 1 and 3, respectively. Sample pages illustrate the diversity in handwriting styles, background types, page quality, and image capture conditions.

Figure 3: Sample word-level images from the Indic-HW-Wild dataset, illustrating diversity in writing styles, illumination conditions, and background noise.

4 Methodology: SemiHastakshar

4.1 Semi-Supervised Fine-Tuning via Pseudo-Labeling

We adopt a semi-supervised learning strategy based on pseudo-labeling [5, 26] to enhance recognition performance for low-resource Indic scripts. The central idea is to iteratively improve the model by leveraging its high-confidence predictions on unlabeled data as additional supervisory signals. The process begins with initializing the model using a checkpoint trained on a limited amount of labeled data. Both labeled and unlabeled handwritten word images are incorporated into the training pipeline. The training proceeds in multiple fixed cycles comprising two stages: **prediction** and **refinement**. The model generates predictions and associated confidence scores for the unlabeled set in each cycle. Only those predictions exceeding a confidence threshold (set at 0.99) are retained as eligible pseudo-labels. From this pool, a subset is randomly sampled such that the number of pseudo-labeled examples equals the number of labeled samples — ensuring a balanced dataset to maintain training stability.

The model is then fine-tuned on the combined set of labeled and pseudo-labeled data for a fixed number of epochs. Early stopping is employed based on validation performance within each cycle to mitigate overfitting. At the end of each cycle, the model checkpoint is updated if it yields improved validation accuracy. The pseudo-labeled set is refreshed using predictions from the newly updated model. This iterative process continues for a predefined number of cycles, progressively enriching the training data and improving generalization without additional human annotations. An overview of this training framework is illustrated in Fig. 4.

4.2 PARSeq for Handwritten Text Recognition

To recognize handwritten Indic text, we leverage PARSeq [7], a Transformer-based model proposed initially for scene text recognition. PARSeq models text sequences using a permuted autoregressive decoding framework, enabling it to generalize beyond fixed left-to-right decoding. It is advantageous when dealing with Indic scripts' varied writing styles and orientations.

The model architecture comprises a Vision Transformer (ViT) encoder and a single-layer Transformer decoder. The encoder divides the input image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ into non-overlapping patches and encodes them as:

$$z = \text{Enc}(x) \in \mathbb{R}^{\frac{W}{p_w} \times \frac{H}{p_h} \times d_{\text{model}}}, \tag{1}$$

where $\mathbf{z}$ is the sequence of visual features.

The decoder applies two attention modules: one attends to previously predicted tokens (context-position attention), and the other attends to image features (image-position attention). These help in modeling dependencies and aligning text with visual context.

The final output is computed by projecting decoder states to character logits:

$$\mathbf{y} = \text{Linear}(\mathbf{h}_{\text{dec}}) \in \mathbb{R}^{(T+1) \times (S+1)}, \tag{2}$$

where S is the size of the character set and h_{dec} is the last decoder hidden state.

A distinctive feature of PARSeq is its use of Permuted Language Modeling (PLM). Instead of learning only from left-to-right decoding orders, PARSeq is trained on multiple random permutations of the output sequence. The expected log-likelihood over such permutations is approximated by:

$$\log p(\mathbf{y}|\mathbf{x}) = \mathbb{E}\mathbf{z} \sim \mathcal{Z}_T \left[\sum_{t=1}^{T} \log p_\theta(y_{z_t} | \mathbf{y}_{\mathbf{z}<t}, \mathbf{x}) \right], \tag{3}$$

where $\mathcal{Z}_T$ denotes a sampled subset of permutations of the output sequence of length T.

This training strategy allows PARSeq to learn more flexible sequence modeling and improves its robustness, especially in low-resource handwritten settings. Its ability to integrate context-aware decoding, iterative refinement, and attention-based alignment makes it a strong baseline for Indic handwritten text recognition.

4.3 Implementation Details

We base all experiments on the PARSeq model, adapting it for Indic handwritten text recognition following the methodology in [7].

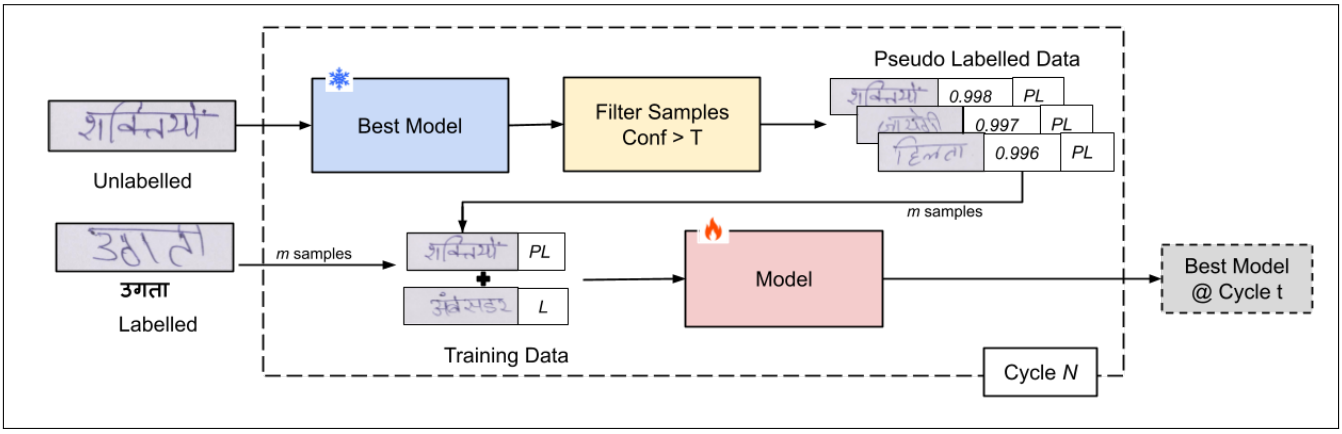

Figure 4: Architecture of SemiHastakshar. Frozen components are denoted by a snowflake icon, and trainable components by a fire icon. The process runs for N cycles, each selecting m samples from both labeled and pseudo-labeled data, with pseudo-label selection governed by a confidence threshold T.

Our approach uses models pre-trained on the labeled subset of the *IIIT-INDIC-HW-WORDS* dataset [7, 18, 25] as the foundation for semi-supervised fine-tuning, detailed in Section 4.

Our model configuration follows the default settings provided in [25], with two key modifications: we increase the dropout rate to 0.4 to enhance generalization and set the learning rate to 1×10^{-7}, which we found to be optimal for stable fine-tuning. The maximum label length is 35 to accommodate longer and more complex word forms that may arise in future extensions. The character set includes language-specific characters, digits, and special symbols. Due to the orthographic and script diversity across Indic languages, the size of the character set varies by language. Experiments are conducted using two NVIDIA GeForce GTX 1080 Ti GPUs.

5 Experiments and Result Analysis

5.1 Datasets

In addition to newly introduced **IIIT-INDIC-HW-WILD** dataset, we also use two additional datasets for our experiments.

IIIT-INDIC-HW-WORDS Dataset: For labeled data, we utilize the *IIIT-INDIC-HW-WORDS* dataset introduced in [18]. This large-scale benchmark comprises approximately 100,000 word-level handwritten images spanning ten major Indic languages: Hindi, Telugu, Bengali, Malayalam, Tamil, Kannada, Oriya, Gujarati, Punjabi, and Urdu. Each word image is annotated with the corresponding transcription, enabling supervised training and evaluation of handwriting recognition models. A detailed summary of the dataset, including the distribution of samples across languages and other characteristics, is provided in Table 2.

IIIT-Indic-HW-UC:. We utilize the *IIIT-Indic-HW-UC* dataset [27], a large-scale corpus of unconstrained handwritten document images spanning thirteen Indic languages. Collected using mobile phone cameras, the dataset captures real-world challenges such as inconsistent lighting, perspective distortion, and background clutter — offering a more realistic benchmark than conventional scanned datasets. The dataset includes contributions from 1,220 writers across various regions of India, introducing substantial diversity in handwriting styles. It contains 91,000 document images, comprising over 2.6M word instances and 566,187 unique words. A detailed breakdown of the dataset statistics is presented in Table 3.

Table 2: Language-wise distribution of writers and word instances in the *IIIT-INDIC-HW-WORDS* dataset.

Script	#Writers	#Word Instances
Hindi	12	95K
Telugu	11	120K
Bengali	24	113K
Gujarati	17	116K
Gurumukhi	22	112K
Kannada	11	103K
Odia	10	101K
Malayalam	27	116K
Tamil	16	103K

Table 3: Language-wise distribution of writers and word instances in the *IIIT-Indic-HW-UC* dataset.

Language	#Writers	#Words Instances
Bengali	158	200K
Gujarati	47	200K
Hindi	118	200K
Kannada	71	200K
Malayalam	137	200K
Oriya	75	200K
Punjabi	67	200K
Tamil	78	200K
Telugu	114	200K

5.2 Training and Evaluation Setup

Our training setup utilizes a combination of labeled and unlabeled data, with labeled samples drawn from the *IIIT-INDIC-HW-WORDS* dataset and unlabeled samples sourced from **Indic-HW-Wild**. As

described in Section 4.1, we employ a semi-supervised training strategy consisting of 7 iterative cycles, each spanning 30 epochs. In each cycle, a new subset of unlabeled data is randomly sampled and pseudo-labeled using a high-confidence threshold of 0.99. Only predictions exceeding this confidence score are retained for further training. For evaluation, we use the *IIIT-INDIC-HW-WORDS* test set to enable direct comparison with prior supervised approaches. Additionally, to assess the generalization ability of our models, we evaluate on the test split of the *IIIT-INDIC-HW-UC* dataset, which serves as an out-of-domain benchmark. We report performance for both the pretrained and fine-tuned models on this dataset.

5.3 Evaluation Metrics

The performance of the proposed model is evaluated using Word Error Rate (WER) and Character Error Rate (CER). WER measures the proportion of incorrectly predicted words relative to the total number of words. A word prediction is considered correct only if all its characters match the ground truth exactly; otherwise, the entire word is marked as incorrect. To capture finer-grained recognition errors, we also report CER, which quantifies discrepancies at the character level. Both WER and CER are computed using the general formula for *Error Rate (ER)*:

$$\text{ER} = \frac{\text{S} + \text{D} + \text{I}}{\text{N}}, \tag{4}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the total number of elements in the reference sequence. For CER, the equation is applied at the character level, whereas for WER, it is applied at the word level.

Table 4: In-domain performance evaluation of PARSeq HTR [25] and SemiHastakshar on the *IIIT-INDIC-HW-WORDS* dataset.

Language	PARSeq HTR [25][a]		SemiHastakshar[b]	
	WER (%)	CER (%)	WER (%)	CER (%)
Hindi	**6.93**	**2.23**	8.87	2.83
Bengali	16.35	4.17	**14.83**	**3.85**
Telugu	**10.37**	2.50	10.38	**2.24**
Gujarati	12.04	2.74	**11.93**	**2.74**
Punjabi	**11.92**	**3.24**	12.46	3.45
Kannada	**6.55**	**1.13**	7.50	1.35
Odia	14.85	**3.36**	**14.48**	3.41
Malayalam	**5.81**	**0.96**	5.99	1.08
Tamil	7.99	1.43	**6.87**	**1.18**

[a] Train Data – *IIIT-INDIC-HW-WORDS (L)*.
[b] Train Data – *IIIT-INDIC-HW-WORDS (L)* + **IIIT-INDIC-HW-WILD (U)**.

5.4 In-domain Result Analysis

Quantitative Analysis: On the **in-domain** *IIIT-INDIC-HW-WORDS* test set, the **SemiHastakshar** achieves performance largely comparable to the baseline. As shown in Table 4, moderate improvements in WER (up to 1.5%) are observed for Bengali, Tamil, and Gujarati, while the baseline PARSeq HTR [25] performs marginally better for Hindi, Kannada, and Malayalam — reflecting its advantage from domain-specific tuning. It indicates that our approach preserves in-domain accuracy despite including diverse pseudo-labeled data.

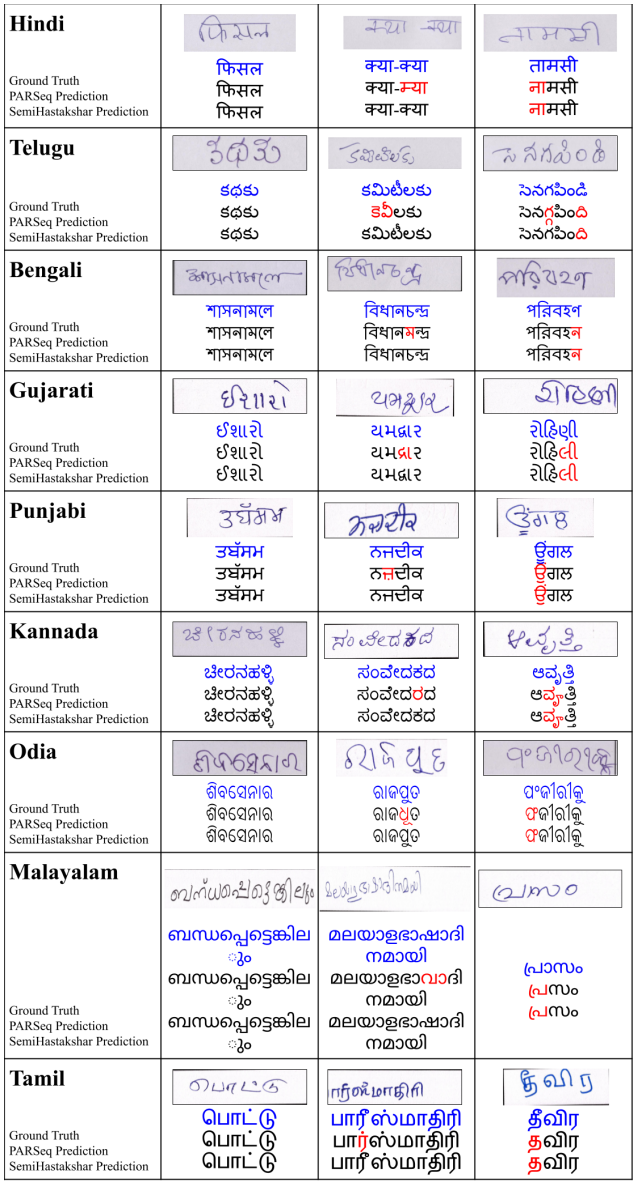

Figure 5: In-domain evaluation on the *IIIT-INDIC-HW-WORDS* dataset for PARSeq HTR [25] and SemiHastakshar. Ground truth text is shown in blue, while incorrectly predicted characters are highlighted in red.

Qualitative Analysis: Visual comparison of some test data samples is shown in Fig. 5. Both models on the in-domain data perform comparatively well. Giving relatively the same errors in characters across languages. It proves that the model does not perform poorly when subjected to pseudo-labeled data during training. It corrects some errors that the earlier PARSeq model trained on

Indic languages failed to recognize correctly, as seen in the second column of Fig. 5.

5.5 Out-of-domain Result Analysis

Quantitative Analysis: The most notable gains appear in the **out-of-domain** setting on the *IIIT-INDIC-HW-UC* test set given in Table 5. **SemiHastakshar** yields substantial WER reductions across all languages — such as 19% for Telugu, 14.4% for Hindi, 12.6% for Bengali, 10% for Tamil, and 7.7% for Gujarati. Corresponding CER drops affirm that improvements extend to character-level recognition. Consistent reductions are achieved even in languages like Punjabi, where the gains are smaller. These results validate the effectiveness of our pseudo-labeling strategy in improving model robustness to real-world, unconstrained handwriting.

Qualitative Analysis: From a qualitative standpoint (Fig. 6), **SemiHastakshar** demonstrates clear improvements over the baseline. It consistently generates outputs that more closely match the ground truth, often correcting major errors observed in the baseline, such as in Tamil samples, and avoiding unnecessary character substitutions. Even in challenging scenarios where complete accuracy is difficult to achieve, **SemiHastakshar** retains partial correctness, whereas the baseline frequently produces entirely incorrect predictions.

Table 5: Out-of-domain evaluation on the *IIIT-INDIC-HW-UC* dataset comparing PARSeq HTR [25] with SemiHastakshar.

Language	PARSeq HTR [25][a]		SemiHastakshar[b]	
	WER (%)	CER (%)	WER (%)	CER (%)
Hindi	53.00	31.94	**38.62**	**26.49**
Bengali	41.71	21.58	**29.09**	**14.86**
Telugu	60.56	30.58	**41.66**	**18.12**
Gujarati	78.39	66.57	**70.70**	**54.73**
Punjabi	43.79	29.32	**43.53**	**28.68**
Kannada	65.06	45.71	**60.53**	**41.18**
Malayalam	59.38	55.43	**55.29**	**44.61**
Tamil	54.99	29.77	**45.02**	**21.13**

[a] Train Data – *IIIT-INDIC-HW-WORDS (L).*
[b] Train Data – *IIIT-INDIC-HW-WORDS (L)* + **IIIT-INDIC-HW-WILD (U)**.

5.6 Dataset-specific Finetuning via Unlabeled Samples

Quantitative Analysis: In cases where handwriting data from a new domain is available but manual annotation is impractical, it becomes essential to explore whether unlabeled samples alone can help build stronger models. We start with the same base model trained on labeled *IIIT-INDIC-HW-WORDS* and apply our semi-supervised training strategy to address this. Unlike earlier experiments, we now incorporate unlabeled samples from the *IIIT-INDIC-HW-UC* dataset alongside the previously used **IIIT-INDIC-HW-WILD** data. The resulting model is referred to as **SemiHastakshar-UC**.

As shown in Table 6, **SemiHastakshar-UC** consistently outperforms the earlier **SemiHastakshar** model across all eight tested

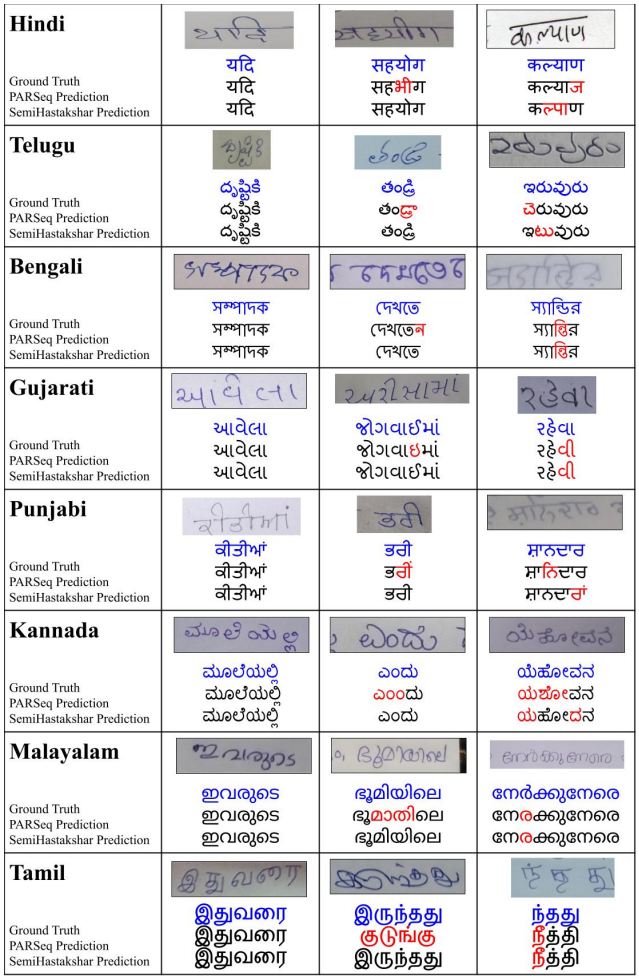

Figure 6: Out-of-domain evaluation on the *IIIT-INDIC-HW-UC* dataset comparing PARSeq HTR [25] with SemiHastakshar. Ground truth text is shown in blue, while incorrectly predicted characters are highlighted in red.

languages. The improvements are substantial. WER drops by 7.89% absolute for Hindi and 12.86% for Punjabi, with corresponding CER reductions of 9.43% and 9.09%, respectively. Tamil also exhibits a notable gain, with WER reducing from 45.02% to 31.69% and CER halving from 21.13% to 13.48%. Bengali, Kannada, and Malayalam show similar trends, achieving WER reductions of 4.79%, 9.05%, and 7.98%, respectively, along with pronounced CER improvements of 3.79%, 8.25%, and 17.56%. These results validate that unannotated data from a target domain, combined with a strong pretrained model, can yield a significantly better model tailored to that dataset. This approach thus reduces the dependence on expensive manual annotation while achieving superior recognition accuracy on the target domain.

Qualitative Analysis: In Fig. 7, we compare the **SemiHastakshar** model to the **SemiHastakshar-UC** model trained on the combined unlabeled dataset of **IIIT-INDIC-HW-WILD** and *IIIT-INDIC-HW-UC*. The first column shows the samples where

both models predict correctly. While the second column shows the samples where the **SemiHastakshar** has failed, **SemiHastakshar-UC** has predicted correctly. A closer comparison of the third column of Fig. 6 and the second column of Fig. 7 shows that the cases where PARSeq HTR [25] and **SemiHastakshar** have failed, **SemiHastakshar-UC** has been able to give correct predictions. It shows that even with the pseudo labels of the *IIIT-INDIC-HW-UC* data, the model can still learn better and give better predictions than the earlier models. The third column displays the sample cases where both models have failed to recognize correctly, showing the scope for further improvement. The cases where both models failed in the given examples are mostly cases where the word samples are distorted, blurred, or poorly segmented. Even in those cases, the models can recognize almost all of the characters, showcasing the models' robust character recognition abilities.

Table 6: Comparison of SemiHastakshar and SemiHastakshar-UC on the *IIIT-INDIC-HW-UC* dataset using only unlabeled domain-specific data for semi-supervised training. Results are reported in terms of Word Error Rate (WER) and Character Error Rate (CER), with the best results for each language in bold.

Language	SemiHastakshar[a]		SemiHastakshar-UC[b]	
	WER (%)	CER (%)	WER (%)	CER (%)
Hindi	38.62	26.49	**30.73**	**17.06**
Bengali	29.09	14.86	**24.30**	**11.07**
Telugu	41.66	18.12	**34.15**	**18.00**
Gujarati	70.70	54.73	**66.90**	**53.84**
Punjabi	43.53	28.68	**30.67**	**19.59**
Kannada	60.53	41.18	**51.48**	**32.93**
Malayalam	55.29	44.61	**47.31**	**27.05**
Tamil	45.02	21.13	**31.69**	**13.48**

[a] Train Data – *IIIT-INDIC-HW-WORDS (L)* + **IIIT-INDIC-HW-WILD (U)**.
[b] Train Data – *IIIT-INDIC-HW-WORDS (L)* + **IIIT-INDIC-HW-WILD (U)** + *IIIT-INDIC-HW-UC (U)*.

5.7 Ablation Study

We conduct an ablation to quantify the effect of labeled data quantity used during supervised pretraining and the semi-supervised fine-tuning on the final model performance. As shown in Table 7, reducing the labeled data from 60k to 30k leads to a consistent and substantial increase in WER and CER across Hindi, Telugu, and Bengali. It suggests sufficient supervised pretraining is crucial for the model to learn robust character representations before leveraging unlabeled data. Despite using the same unlabeled data, the deterioration in accuracy indicates that semi-supervised fine-tuning alone cannot compensate for limited supervision in the initial training phase.

Limitations: Even though our framework demonstrates significant potential of semi-supervised learning for Indic HTR, it faces several limitations. The model's performance depends on the quality of the pseudo-labels. If the model confidently makes an incorrect prediction, this "noisy" label is incorporated into the training data leading to confirmation bias. Another key limitation is the reliance on the high-confidence threshold which results in systematic neglect of informative samples that fall below this threshold.

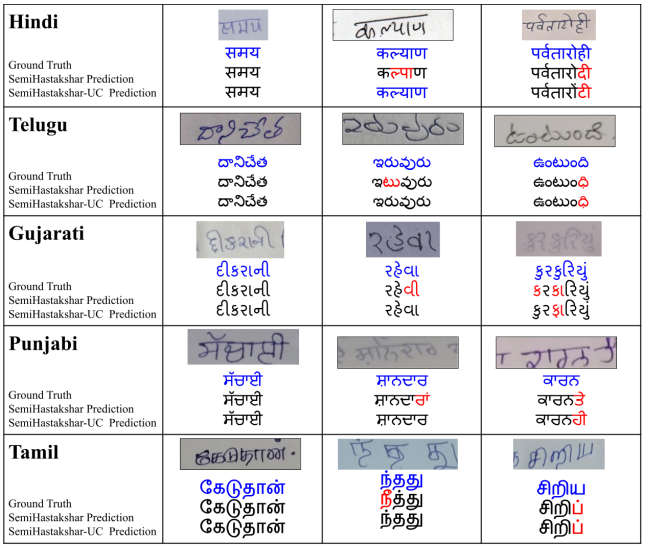

Figure 7: Qualitative comparison between SemiHastakshar and SemiHastakshar-UC on the *IIIT-INDIC-HW-UC* dataset. Text in blue denotes the ground truth, while characters highlighted in red indicate prediction errors.

Table 7: Impact of labeled data size on semi-supervised training performance for the *IIIT-INDIC-HW-UC* dataset.

Language	Full Data (60k–80k)[a]		30k Samples[b]	
	WER (%)	CER (%)	WER (%)	CER (%)
Hindi	**38.62**	**26.49**	57.89	58.73
Telugu	**41.66**	**18.12**	66.16	54.29
Bengali	**29.09**	**14.86**	41.43	22.87

[a] Train Data: *IIIT-INDIC-HW-WORDS (L, 60k–80k)* + **IIIT-INDIC-HW-WILD (U)**
[b] Train Data: *IIIT-INDIC-HW-WORDS (L, 30k)* + **IIIT-INDIC-HW-WILD (U)**

6 Conclusion

We presented a semi-supervised framework to address generalization challenges in Indic handwritten text recognition. We fine-tuned a PARSeq model to learn from diverse handwriting styles by leveraging large-scale unlabeled data through high-confidence pseudo-labeling. Our approach **SemiHastakshar** significantly improved out-of-domain performance on the *IIIT-INDIC-HW-UC* dataset, while maintaining competitive results on the in-domain test set. It demonstrates the model's improved robustness and generalization. The proposed method offers a scalable solution for low-resource scripts and highlights the untapped potential of unlabeled data in document image analysis. Future work includes exploring advanced semi-supervised techniques.

Acknowledgments

This work is supported by MeitY, Government of India, through the NLTM-Bhashini project.

References

[1] Chandranath Adak, Bidyut B Chaudhuri, and Michael Blumenstein. 2016. Offline cursive Bengali word recognition using CNNs with a recurrent model. In *2016 15th International conference on frontiers in handwriting recognition (ICFHR)*. IEEE, 429–434.

[2] Juan Manuel Alonso-Weber, MP Sesmero, and Araceli Sanchis. 2014. Combining additive input noise annealing and pattern transformations for improved handwritten character recognition. *Expert systems with applications* 41, 18 (2014), 8180–8188.

[3] Chris Andrew, Santhoshini Reddy, Viswanath Pulabaigari, and Umapada Pal. 2017. Text independent writer identification for Telugu script using directional filter based features. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 5. IEEE, 65–70.

[4] Sandhya Arora, Debotosh Bhattacharjee, Mita Nasipuri, Latesh Malik, Mohantapash Kundu, and Dipak Kumar Basu. 2010. Performance comparison of SVM and ANN for handwritten devnagari character recognition. *arXiv preprint arXiv:1006.5902* (2010).

[5] Jeonghun Baek, Yusuke Matsui, and Kiyoharu Aizawa. 2021. What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. N/A, 3113–3122.

[6] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. 2019. Character region awareness for text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9365–9374.

[7] Darwin Bautista and Rowel Atienza. 2022. Scene text recognition with permuted autoregressive sequence models. In *European conference on computer vision*. Springer, 178–196.

[8] Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*. 92–100.

[9] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. 2020. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems* 33 (2020), 22243–22255.

[10] Remi Denton, Sam Gross, and Rob Fergus. 2016. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv:1611.06430* (2016).

[11] W Dong-Dong Chen and ZH Wei Gao. 2018. Tri-net for semi-supervised deep learning. In *Proceedings of twenty-seventh international joint conference on artificial intelligence*. 2014–2020.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[13] Kartik Dutta, Praveen Krishnan, Minesh Mathew, and CV Jawahar. 2017. Towards accurate handwritten word recognition for Hindi and Bangla. In *National Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics*. Springer, 470–480.

[14] Kartik Dutta, Praveen Krishnan, Minesh Mathew, and CV Jawahar. 2018. Towards spotting and recognition of handwritten words in Indic scripts. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 32–37.

[15] Utpal Garain, Luc Mioulet, Bidyut B Chaudhuri, Clement Chatelain, and Thierry Paquet. 2015. Unconstrained Bengali handwriting recognition with recurrent models. In *2015 13th international conference on document analysis and recognition (ICDAR)*. IEEE, 1056–1060.

[16] Carlos Garrido-Munoz and Jorge Calvo-Zaragoza. 2025. On the generalization of handwritten text recognition models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 15275–15286.

[17] Ayanabha Ghosh. 2023. Towards Full-page Offline Bangla Handwritten Text Recognition using Image-to-Sequence Architecture. In *2023 IEEE Silchar Subsection Conference (SILCON)*. IEEE, 1–6.

[18] Santhoshini Gongidi and CV Jawahar. 2021. iiit-indic-hw-words: A Dataset for Indic Handwritten Text Recognition. In *International Conference on Document Analysis and Recognition*. Springer, 444–459.

[19] Government of India. 1950. The Constitution of India. Eighth Schedule – Languages. https://legislative.gov.in/sites/default/files/COI.pdf. Ministry of Law and Justice. Retrieved July 28, 2025.

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

[21] Kylie Jones and Carolyne Bird. 2025. Complexity, Features, and Comparisons in Forensic Handwriting Examination. *Wiley Interdisciplinary Reviews: Forensic Science* 7, 1 (2025), e1537.

[22] Harmandeep Kaur and Munish Kumar. 2021. On the recognition of offline handwritten word using holistic approach and AdaBoost methodology. *Multimedia Tools and Applications* 80, 7 (2021), 11155–11175.

[23] Mahdieh Labani, Parham Moradi, Fardin Ahmadizar, and Mahdi Jalili. 2018. A novel multivariate filter method for feature selection in text classification problems. *Engineering Applications of Artificial Intelligence* 70 (2018), 25–37.

[24] Samuli Laine and Timo Aila. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242* (2016).

[25] Evani Lalitha, Ajoy Mondal, and CV Jawahar. 2024. Enhancing Accuracy in Indic Handwritten Text Recognition. In *International Conference on Computer Vision and Image Processing*. Springer, 234–248.

[26] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, Vol. 3. Atlanta, 896.

[27] Ajoy Mondal and CV Jawahar. 2024. Unconstrained Camera Captured Indic Offline Handwritten Dataset. In *International Conference on Pattern Recognition*. Springer, 333–348.

[28] Sk Md Obaidullah, Chayan Halder, KC Santosh, Nibaran Das, and Kaushik Roy. 2018. PHDIndic_11: page-level handwritten document image dataset of 11 official Indic scripts for script identification. *Multimedia Tools and Applications* 77, 2 (2018), 1643–1678.

[29] Augustus Odena. 2016. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583* (2016).

[30] Abu Quwsar Ohi. 2020. BanglaWriting: A multi-purpose offline Bangla handwriting dataset (script).

[31] Sarah Palmis, Jeremy Danna, Jean-Luc Velay, and Marieke Longcamp. 2019. Motor control of handwriting in the developing brain: A review. *Developmental Dysgraphia* (2019), 123–140.

[32] Md Ataur Rahman, Nazifa Tabassum, Mitu Paul, Riya Pal, and Mohammad Khairul Islam. 2023. Bn-htrd: A benchmark dataset for document level offline bangla handwritten text recognition (htr) and line segmentation. In *Computer Vision and Image Analysis for Industry 4.0*. Chapman and Hall/CRC, 1–16.

[33] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. 2015. Semi-supervised learning with ladder networks. *Advances in neural information processing systems* 28 (2015).

[34] Arshia Rehman, Saeeda Naz, and Muhammad Imran Razzak. 2019. Writer identification using machine learning approaches: a comprehensive review. *Multimedia Tools and Applications* 78, 8 (2019), 10889–10931.

[35] Partha Pratim Roy, Ayan Kumar Bhunia, Ayan Das, Prasenjit Dey, and Umapada Pal. 2016. HMM-based Indic handwritten word recognition using zone segmentation. *Pattern recognition* 60 (2016), 1057–1075.

[36] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems* 29 (2016).

[37] Naveen Sankaran, Aman Neelappa, and CV Jawahar. 2013. Devanagari text recognition: A transcription based formulation. In *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 678–682.

[38] Monika Sethi, Munish Kumar, and MK Jindal. 2025. Forensic handwriting analysis: a hybrid classification framework for writer identification in Devanagari script. *Multimedia Tools and Applications* (2025), 1–12.

[39] Bikash Shaw, Ujjwal Bhattacharya, and Swapan K Parui. 2014. Combination of features for efficient recognition of offline handwritten Devanagari words. In *2014 14th International conference on frontiers in handwriting recognition*. IEEE, 240–245.

[40] Bikash Shaw, Swapan Kumar Parui, and Malayappan Shridhar. 2008. Offline Handwritten Devanagari Word Recognition: A holistic approach based on directional chain code feature and HMM. In *2008 International Conference on Information Technology*. IEEE, 203–208.

[41] Jost Tobias Springenberg. 2015. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390* (2015).

[42] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems* 30 (2017).

[43] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. 2022. A survey on deep semi-supervised learning. *IEEE transactions on knowledge and data engineering* 35, 9 (2022), 8934–8954.