Beyond Script Family Boundaries: Towards Unified Open-Set Scene Text Recognition

Chang Liu and Elisa H Barney Smith

Machine Learning Group, Luleå Technical University, Sweden
{chang.liu,elisa.barney}@ltu.se

Abstract. Although VLMs are making fast progress in recognizing non-Latin scripts, they still face feasibility problems for less common scripts with extremely low resources. Current Open-set text recognition methods show some good qualitative results, although they still lack quantitative evidence. Specifically, the majority of open-set and zero-shot text recognition methods are confined to the CJK family, and to our knowledge, none handle more than one script family at the same time. In this work, we fill this gap with a unified, script-agnostic open-set text recognition framework that can handle not just multiple scripts, but multiple script families - Indic, Latin, and CJK scripts - in one set of partially shared modules. In this framework, we introduce a robust network structure that can handle the complex character layout across a diverse collection of scripts of different families. Furthermore, we introduce a heterogeneous co-training method and tasks to enhance the model performance on unseen scripts, and discuss the impacts from different data sources and task formats (character and word recognition). Experiments demonstrate that the proposed framework shows zero-shot learning capability on natural scene word images of Bengali, Gujarati, Japanese, Korean, and synthetic Yi samples. Specifically on the rare, yet actively used, Yi script, our framework can achieve a word accuracy (ACR) of 16.10%, while the GPT5.2 shows an ACR of only 0, indicating a fundamental capability gap.

1 Introduction

Although modern vision language models (VLMs) like Gemini and ChatGPT are able to recognize the content of documents and scene text images written in English and Chinese [31], for relatively minor scripts with a vast and diverse character set, they still face fundamental capability problems in transcribing scene text images (see Fig. 1). Commercial VLMs do have huge token sets that cover minority scripts like the Yi script, but they still fail in the task of comparing specific shapes of individual glyphs, yielding complete hallucination. This indicates the VLMs fail to model either "to be" (the concept of being the same character) or "not to be" (the concept of being different characters).

Zero-shot text/character recognition methods [47] offer the capability to recognize unseen characters by matching parts from the characters. This has been

Fig. 1: Limitation of commercial VLMs on recognizing minor scripts

used considerably for Chinese-Japanese-Korean (CJK) text recognition research, where the number of characters in the scripts is orders of magnitude greater than European scripts. However, most such methods [37, 42, 47] require script-specific knowledge like radical [4], stroke composition [9], other structural representations [16], or a combination of the above [42, 47] for each character category, limiting them to the CJK family. Furthermore, they cannot spot "unknown unknown" [12] characters from the datastream and thus are prone to what is commonly called "hallucinations". Although the rejection capability has been implemented in recent open-set text recognition methods [21, 24], they only report quantitative results on Japanese and Korean, indicating limited generalization capability and scalability beyond CJK and Latin scripts. To improve generalization capability, recent zero-shot [44] and open-set [23] text recognition methods pretrain or co-train with individual character glyphs, yet their sources remain monolithic. Beyond the CJK family, a handful of methods [6, 7] work on Bengali and Latin using script-specific knowledge. Yet, to our knowledge, there does not exist a proven, script-agnostic open-set text recognition method that handles multiple script families within one model.

The above limitation inspires the following research questions, which we try to answer in this work:

- RQ1. Can we scale open-set text recognition beyond CJK?
- RQ2. How can we improve performance through co-training?
- RQ3. Does human perception of "to be" generalize across scripts?

To answer these research questions, we first propose a scalable heterogeneous multi-task joint co-training framework that allows us to co-train an open-set text recognition task alongside an arbitrary number of other open-set text recognition and character classification tasks.

We then systematically investigate a diverse collection of data sources and co-training tasks. For word recognition, we include (1) simplified Chinese datasets [24], (2) the Indic street text dataset [30], (3) word-level synthetic data, and (4) the MJ and ST dataset of the ABINet variant [11]. For character recognition, we use dynamically generated synthetic character data similar to CFOR [23].

For benchmarking, we adopt the open-set text recognition benchmarking protocol [24] for the sake of referenced comparison, and expand it with a generalized zero-shot benchmarks on Bengali and Gujarati, and a benchmark on synthetic Yi-script words.

In summary, the contributions of this work are:

- A script family agnostic, multi-task co-training framework that is robust against inter-script-family and inter-script conflicts.
- To our knowledge, the first Scene Text Recognition framework that demonstrates zero-shot recognition capability on unseen Indic, CJK, and Yi script families at the same time.
- Achieving comparable generalization capability on zero-shot Korean and Japanese OCR performance against SOTA MoE models.

2 Related Works

2.1 Multitask Co-training

Co-training has been used since 1997 [5], and has been applied to many tasks [35]. One common way to do this is by pretraining the model on a larger dataset like ImageNet [10] and large-scale image caption pairs [32]. The multi-task learning, or co-training paradigm emerged [33] and demonstrated scalability and generalization capacities against novel tasks [13]. This property leads to foundation models that convert various tasks to homogeneous formulations and train them together [20, 39, 40]. Aside from pretraining [45], co-training is actively used in text recognition. Cross-language co-training is seen as an approach to improve word or line-level recognition performance [2, 28].

Conflicts are sometimes noticed when datasets are size-limited and are often addressed by training strategies, e.g., balanced finetuning and mixture of experts [17, 46]. There is an emerging trend of adapting open source foundation models for document analysis, like DeepseekOCRv2 [38], which showcases both the pretraining by being a Qwen-vl adaptation [36] and also demonstrates co-training patterns by co-training downstream tasks like transcription and layout analysis together, yet co-training with heterogeneous tasks has not been highly adopted in the field of scene text recognition.

Although current developments with foundation models for OCR show strong performance with in-distribution data [31], current commercial VLMs still demonstrate a limitation on data scarce tasks like Yi OCR showcased in Fig 1, causing an inequity and exclusion for minor scripts and users, leaving OCR with Unknown Characters as an open field for researchers.

2.2 OCR with Unknown Characters

Currently, two major capabilities are involved to address unseen characters: rejection and recognition. When considering the presence of new, formerly unseen characters, the majority of the OCR community focuses on the recognition side, aiming to recognize the new characters without retraining, formulating the problem as few-shot [37] or zero-shot [4] learning. This community has mostly researched Chinese, Japanese, and Korean (CJK) character recognition [9, 37, 42, 47] due to its uniquely large character set [44] and vast data

availability, with some works on other script families like Latin [34], Bengali [7], and Oracle [27]. There are two primary ways to recognize unseen Chinese characters: by matching components [4,37] and matching glyphs [1,25]. Some methods like [9] use a combination of both approaches. Some recent approaches have moved beyond recognizing individual characters to recognizing words or text lines [18, 25, 43] and have demonstrated some extent of close-set recognition performance, indicating a certain level of real-world feasibility. However, these methods only address the recognition problem after novel characters are spotted from the data stream; they cannot actively spot novel characters that do not have registered side information, leading to silent recognition errors. The above limitation leads to demand for rejection capability, known as open-set recognition (OSR) [41].

The OSR formulation is adapted to the problem of text recognition as Open-Set Text Recognition (OSTR) [24], which demands both recognition and rejection capabilities. The open-set text recognition task formulation decouples action (recognition and rejection) from the "novelty" of the label (seen or novel (unseen) in the training samples), and ties action to the presence of side-information (glyphs, strokes [9], radical [37], or other forms of side-information [16]), where categories with side-information registered are considered "in-set" and subject to recognition, and are otherwise considered as "out-of-set" and subject to rejection. Although recent methods demonstrate reasonable generalization capability from simplified Chinese to multi-orientation Japanese [22] and even Korean [21], such methods are still confined to the Latin and CJK families despite their claim of being script agnostic.

Although a recent OSTR method [23] co-trains the method with glyphs from the entire Noto font collection, their evaluation is still limited to the Latin and CJK families. In this work, we propose a unified open-set text recognition framework, which demonstrates various extents of generalization capabilities on several scripts and script families at the same time. Specifically, the proposed framework shows inductive zero-shot recognition capabilities on Japanese, Korean, and Yi, and transductive zero-shot recognition capabilities on Bengali and Gujarati.

3 Method

In this work, we propose a scalable, modular framework (Fig. 2 (top)) to address the diversity of characters and layouts across multiple script families, which is composed of multiple heterogeneous open-set text recognition tasks sharing a common pool of modules.

3.1 Tasks

The framework involves two types of tasks, namely the character recognition task and the word recognition task (shown in Fig. 2 (bottom)). A task is defined as modules and dataflows organized to recognize words of a script group sg (e.g. CJK, Indic, etc.). Specifically, a task is composed of a set of one or more

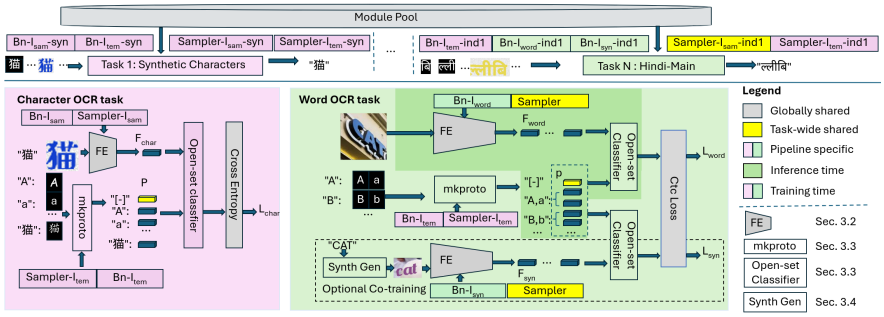

Fig. 2: The proposed unified Open-Set text recognition framework (top), the implementation of a Character-based task (bottom left), and the implementation of a word-based task (bottom right).

Region of Interest (RoI) Feature Extractors $\texttt{FE}[0], ...\texttt{FE}[N]$ (Section 3.2), one dedicated prototyping module `mkproto`, corresponding Open-Set Classifier `OSC` (Section 3.3), and optimization objectives.

Within the scope of the main paper, the character recognition task is only involved in training as a synthetic task and only has one RoI Feature Extraction[1]. For each batch, it samples 64 character categories, and for each character category, generates two sample images and one side information using the Character Data Generator (see Sec 3.4). It aligns features extracted from the sample image with prototypes generated from side information with a cross entropy loss:

$$L_{char} = \texttt{CrossEntropy}(\texttt{OSC}(\texttt{FE}(I_{sam}, BN^{sg}, \texttt{S}^{sg}_{char}), \texttt{mkproto}(I_{tem})), y). \quad (1)$$

Like the character recognition task, the word recognition tasks always include one RoI Feature Extraction for real samples, one prototyping module `mkproto` for prototype generation, and one dedicated Open-Set Classifier `OSC` for class boundary modeling. However, the word recognition tasks are involved in both training and testing, and can include an additional optional dataflow for synthetic co-training data. During training, the task samples [24] a set of side information, and encodes it to p_{batch}:

$$I_{tem} = \texttt{Sample}(I_{tem}, y), p_{batch} = \texttt{mkproto}(I_{tem}). \quad (2)$$

Instead of using the position-wise decoder and loss like [24, 25] for the word recognition task, we propose a CAM-CTC-based structure, trained with word level `CTC` loss:

$$L_{word} = \texttt{CTC}(\texttt{OSC}(\texttt{FE}(I_{sam}, BN^{sg}, \texttt{S}^{sg}_{seq}), p_{batch}), y). \quad (3)$$

We found that this structure is much more robust within and beyond the CJK family (quantitative evidence can be found in Section 4.2). We attribute this to

[1] It is applied for Oracle characters recognition in the supplementary materials.

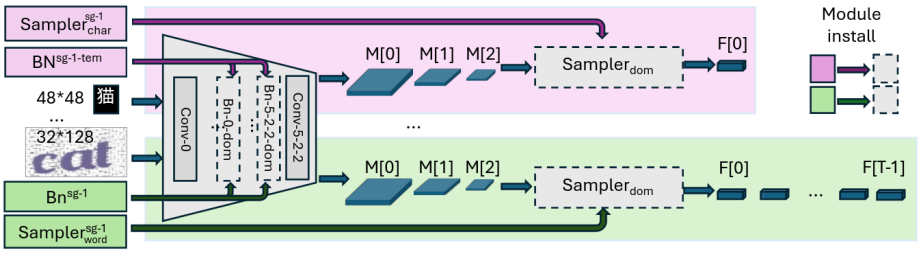

Fig. 3: RoI Feature Extraction for character data (magenta) and word data (green).

the difficulty in separating characters from each other in unseen scripts, which is alleviated by the CTC loss [14], allowing sampler $\mathtt{S}^{sg}_{seq}$ to be less rigid.

If the task is configured with co-training, the input labels are sent to the Word Data Generator (detailed in Sec 3.4) to generate synthetic samples I_{synw} and corresponding labels y_{synw}, trained with synthetic sample `CTC` loss:

$$L_{syn} = \mathtt{CTC}(\mathtt{OSC}(\mathtt{FE}(I_{synw}, BN^{sg}_{syn}, \mathtt{S}^{sg}_{seq}), p_{batch}), y_{synw}). \tag{4}$$

Note, the sampler $\mathtt{S}^{sg}_{seq}$ is shared with real data, but not the batch normalization.

During test, the modules in `mkproto` are frozen, thus p can be cached before inference, reducing the footprint to only the Open-Set Classifier and the RoI Feature Extraction (highlighted in the darker green shade in Fig. 2).

3.2 Region of Interest (RoI) Feature Extraction

In this work, we formulate the process to extract character representation(s) from images I_{dom}(characters, words, glyphs) as a single process of Region of Interest (RoI) Feature Extraction `FE` (illustrated in Fig. 3) :

$$F_{dom} = \mathtt{FE}(I_{dom}, BN_{dom}, \mathtt{S}_{dom}). \tag{5}$$

The feature sequence $F : \mathcal{R}^{maxT \times d}$ is the extracted representation of characters, where $maxT$ is the length of the sequence and d is the dimension of the character representation. BN_{dom} and $\mathtt{S}_{dom}$ are the batch normalization layers and the Sampler module tied to the domain (dom) of the input image [8]. The process can be further broken down into two sub-processes: namely partially shared image feature extractor `Enc` and RoI sampling with the input Sampler `S`.

In the partially shared image feature extractor, the module first inserts the input batch normalization layers after the shared convolution layers (the dashed boxes) in the backbone. It then extracts the feature map tower M with the assembled convolution backbone:

$$M = \mathtt{Enc}(I_{dom}, BN_{dom}). \tag{6}$$

In the RoI sampling, the model uses the input sampler to compute the RoI mask A of each RoI, and samples them from the corresponding locations on the

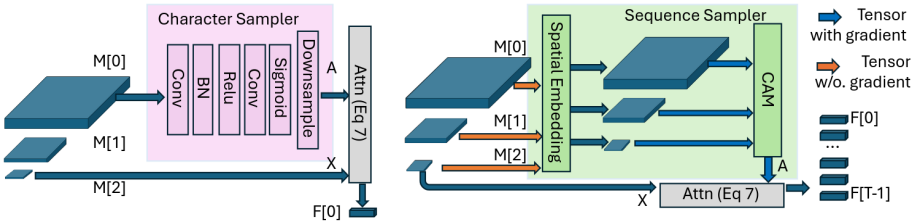

Fig. 4: The implementation of the character feature sampler (left) and the word feature sampler (right).

last feature map $M[2]$:

$$F_{dom}[t] = \frac{A[t] * M[2]}{\sum A[t]}, \text{where } A = \mathtt{S}_{dom}(M). \tag{7}$$

Here, $t \in \{0, 1, ..., |A| - 1\}$ is the timestamp of the sequence.

In this work, we use two implementations (Fig. 4) of sampler processes $\mathtt{S}_{dom}$ for sampling individual characters ($\mathtt{S}_{char}$) and sequences of characters in words ($\mathtt{S}_{seq}$). Both implementations of $\mathtt{S}_{dom}$ take the feature map tower M as input, and output the attention mask sequences A.

For character data, we use a simple local pattern-based attention sampler by using a shallow convolution block with a single output channel to estimate the foreground, i.e. RoI mask A, from $M[0]$:

$$A = \mathtt{S}_{char}(M) = \mathtt{Downsample}(\mathtt{Convblk}(M[0])). \tag{8}$$

For word data, we adapt the Convolutional Attention Module (CAM) module from OpenCCD [25]. Specifically, it first detaches the gradient of the feature map tower and adds spatial embedding. The embedded features are then fed into the CAM:

$$A = \mathtt{S}_{seq}(M) = \mathtt{CAM}(\mathtt{SpatialEmb}(\mathtt{detach}(M))). \tag{9}$$

Though implementation-wise this is similar to OpenCCD [25], there is no longer a need to predict the sequence length due to the CTC loss and the beam-search based decoder used in the tasks. Consequently, the RoI masks A, and the feature sequence F are no longer strictly aligned to the ground truth.

3.3 Open-Set Recognition

While the synthetic co-training augments the label, context, and style space for more robust representations, the core capability of Open-Set recognition comes from the prototyping module `mkproto` and the Open-Set Classifier `OSC`.

Prototyping Module A prototyping module `mkproto` (Fig. 5 left) enables the model to convert a set of glyph images[2] into class centers used by the Open-Set

[2] or exemplars for single shot learning oracle character recognition, see Supp. Mat..

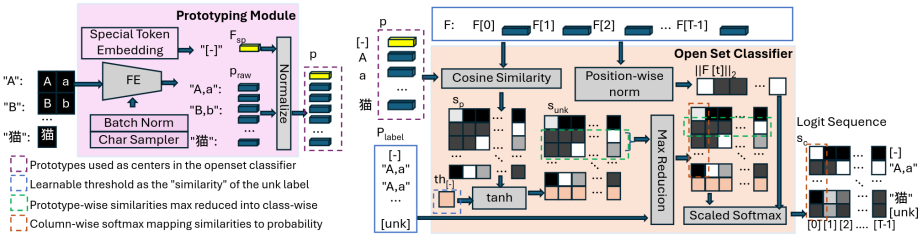

Fig. 5: The Prototyping Module `mkproto` (left) and the Open-Set Classifier `OSC`(right).

Classifier for a specific script group, enabling dynamic novel categories injection without retraining the model.

The prototyping module holds a dedicated character sampler $\mathtt{S}^{sg}_{char}$, a set of Batch Normalization layers BN^{sg}_{tem}, and script-group-specific special token embeddings E^{sg}_{sp} for the blank character [14]. The module takes side information I_{tem} as input and produces class centers (prototypes) p:

$$p = \mathtt{mkproto}(I_{tem}). \tag{10}$$

Specifically, the module first encodes the glyph images of normal tokens (graphemes) via a RoI Feature Extraction. The raw prototypes p_{raw} are then appended behind the special token embeddings E_{sp}, which are trainable embeddings for shape-less control characters (tokens) like "[-]" used in the CTC loss [14], which is consequently normalized by the L2-Norm [24].

In summary, `mkproto` can be defined as:

$$p_{raw} = \mathtt{FE}(I_{tem}, BN^{sg}_{tem}, \mathtt{S}^{sg}_{char}), p = \mathtt{normalize}([E^{sg}_{sp}, p_{raw})]). \tag{11}$$

Open-Set Classifier The Open-Set Classifier `OSC` (Fig.5, right), on the other hand, implements the rejection capability that allows the model to predict a character representation $F[i]$ as the "[unk]" label. The module takes feature sequence F, prototypes p, and support label y_t as inputs, and produces the logit sequence s_c:

$$s_c = \mathtt{OSC}(F, p, y_t). \tag{12}$$

Here, y_t is the label of p. Note that more than one prototype in p, e.g. prototypes for 'a' and 'A', can be mapped to the same label in y_t.

The module holds a single learnable unknown threshold $th_{[-]}$ as the similarity threshold for assigning the "[unk]" label. Specifically, the module first computes the cosine similarity between p and each timestamp in F. Then the model appends the hyperbolic tangent of the learnable unknown threshold as the "similarity" for the "[unk]" label in the sub-category scores:

$$s_p[i,j] = \mathtt{cosine}(F[i], p[j]), s_{unk}[i] = [s_p[i], \mathtt{tanh}(th_{[-]})]. \tag{13}$$

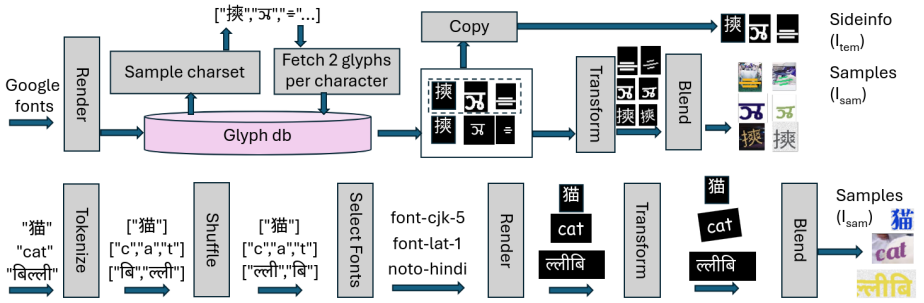

Fig. 6: Heterogeneous Synthetic Data Generator. (top) Character Data Generator - All supported glyphs are rendered and added to a database. Glyphs are used as side information. (bottom) Word Data Generator - Sample words are broken into tokens, reordered, and re-rendered to increase the data quantity and variety.

Finally, a max reduction is applied to merge the prototype-wise similarity s_{unk} into class-wise similarity, which is scaled into class-wise logits:

$$s_c[t, i] = ||F[t]||_2 \max_{y_t[j]=i} s_{unk}[t, j]. \tag{14}$$

Here, $||F[t]||_2$ is the L_2 norm of the sample feature vector at location t[3]. Note, the norm does not change the "ranking" of the similarity between the prototypes and a certain feature vector $F[t]$. Instead, it serves as the temperature, which affects the "one-hot-ness" of the softmax output.

3.4 Heterogeneous Synthetic Data Generator

In this work, we propose to co-train the framework with heterogeneous synthetic data on both the character level and word level. The generation processes are illustrated in Fig. 6.

Character Data Generator For character-level data generation, the generator first renders all supported glyphs into binary masks and caches them in a database. It then removes blank glyphs and identical glyphs shared by more than 5 different UTF characters; these shapes are usually the placeholder for missing glyphs. During training, the generator first samples a subset of all characters, then fetches 2 glyphs for each selected character. The first glyph mask of each character is used as its side information I_{tem}. All glyph masks are then augmented and blended with the background as sample images I_{sam}.

Word Data Generator For word-level data generation, the generator reuses a corpus from real word samples, to make sure characters in each generated sample are likely from one script so they can be covered by a single font. The generator

[3] It's not necessarily aligned to the t-th character due to the CTC loss

first tokenizes each ground truth word into individual Unicode graphemes, and then shuffles the graphemes within each word as the labels for the synthetic data. For each shuffled word, the generator samples a font that supports all characters for each label, and renders it into a binary mask. The binary masks are then augmented with geometrical transformations and blended on randomly generated backgrounds as sample images I_{sam}.

4 Experiments

In this section, we first go through the ablative studies over network structure and choices of synthetic co-training tasks in Section 4.2. Then, we benchmark the zero-shot text recognition capability (to recognize novel characters and scripts) in Section 4.3. Finally, we benchmark the open-set text recognition capability (to reject samples with unregistered characters) in Section 4.4. Close-set experiments and single-shot oracle character recognition experiments are in the supplementary materials.

4.1 Implementation Details

Framework Configurations In this work, we train three different configurations of frameworks: one with only Simplified Chinese and synthetic data (Ours-CJK), one with only Indic-related and synthetic tasks (Ours-IND), and a unified configuration (Ours-Uni) that combines all tasks.

Specifically, configuration Ours-CJK is composed of the Synthetic Characters task and the CJK-Main, which is a word recognition task training on simplified Chinese datasets [24] with co-training enabled. This configuration also includes an additional synthetic Yi script dataset for testing.

For the Ours-IND configuration, we train on 4 different tasks, namely Synthetic Characters, Punjabi, Tamil, and Indic-Main, on a combination of Hindi, Malayalam, Kannada, Marathi, Telugu, and Oriya scripts. The split within the Indic family is to address the inter-script conflict discussed in Section 5.1. Due to computational resource limitations, we do not implement synthetic co-training for the standalone Punjabi and Tamil tasks.

For Ours-Uni, we combine the CJK and Indic configurations, and add the IIIT5k task where the model is trained on the MJ [19] and ST [15] dataset, and tested on the IIIT5k [29] testing task, resulting in a configuration of 6 tasks in total: Synthetic Characters, CJK-Main, Punjabi, Tamil, Indic-Main, and IIIT5k.

Implementation and Hyperparameters The implementation is built on the WnA codebase [21] with the routing system removed for implementation simplicity, where specific network structures can be found in the GitHub repository. All models are trained with $140,000$ iterations. Models, code, data, and documents can be found on `https://github.com/lancercat/uni-zero`.

Table 1: Comparison of the proposed Word Pipeline structure. Models for Japanese, Korean, and Yi are trained on Simplified Chinese data, while Models for Bengali and Gujarati are trained on Indic datasets without these two scripts. Results are shown in ACR (Avg. ↑ / Std. ↓)

	OpenCCD-like [25]	Rosetta-like [3]	**Ours**
Sampling	CAM	Squeeze	CAM
Alignment	Position-wise	CTC	CTC
ACR (JP)	41.59/0.19	41.79/0.30	**42.36**/0.48
ACR (KR)	15.62/1.41	12.65/1.32	**19.18**/1.08
ACR (Yi)	5.73/5.73	6.68/0.82	**8.51**/1.05
ACR (Bengali)	2.81/2.81	3.36/1.44	**4.74**/0.17
ACR (Gujarati)	2.30/2.30	5.17/0.62	**5.98**/0.55

Synthetic Yi Dataset The Yi dataset contains 5,000 samples with lengths ranging from 1 to 10. Specifically, for each length l in $[1, 10]$, we generate 500 random samples with ID ranging from $[(l-1)*500, l*500)$. For each random sample of length l, we randomly draw l characters from the character set and render it into an image using the synthetic engine in this paper. To save costs for LLM testing, we downsampled the dataset by taking out 50 samples whose ID mod 100 = 0. The sampled dataset has 5 samples for each length.

4.2 Ablative Studies

In this work, we first conduct ablative studies on the network structure, then discuss the effect of a combination of different synthetic tasks. We conduct ablative studies on the Generalized Zero Shot Learning (GZSL) setup following existing works [21, 24].

Word Pipeline structure In this part, we compare the impact of three different alignment designs, namely the Rosetta-like CNN-Squeeze-CTC [3] (alignment by CTC), the OpenCCD-like CNN-CAM-PositionWise [25] (Alignment by attention), and the proposed CNN-CAM-CTC structure (alignment by both).

Quantitative results are shown in Table 1. Results show that depending only on CTC or attention for alignment yields noticeably worse performance compared to using both for the alignment, indicating the proposed method is more robust against various scripts and spatial layouts.

This set of experiments suggests that feature alignment and sampling are very important parts of open-set text recognition, where sampling robustness is directly tied to the capability to recognize novel characters.

Synthetic Co-training In this part, we discuss how different co-training approaches and their combinations affect the GZSL setup performance on different scripts. Results in Table 2 show that co-training with synthetic word data (LSCT-Word) can yield around 2% performance gain for the CJK script family.

Table 2: Comparison of Synthetic Co-training approaches in ACR (Avg. ↑/ Std. ↓)

	No Co-training	2X Batch size	LSCT-Word	LSCT-Char	**LSCT-Full**
Word	No	No	Yes	No	Yes
Character	No	No	No	Yes	Yes
ACR (JP)	42.36/0.48	43.39/0.25	44.24/0.46	45.17/0.26	**46.08**/0.29
ACR (KR)	19.18/1.08	16.60/0.55	21.87/0.60	23.30/0.34	**23.50**/0.49
ACR (Yi)	8.51/1.05	7.49/0.66	11.42/0.54	12.05/0.31	**17.07**/1.45

Adding character-level co-training further improves the recognition performance on the unseen script by 2% or more. While co-training only synthetic characters (LSCT-Char) is more effective than word-level co-training, it still lags behind the combination of the two heterogeneous co-training tasks.

In addition, to determine whether it was the co-training itself or the addition to number of samples that caused the improvement, we doubled the batch size of the base model, to match the total number of samples in LSCT-Word. Although this improved the performance on Japanese a bit, where there are seen characters, it leads to noticeable degradation on Yi and Korean, where all characters are novel to the model. On the other hand, the LSCT-Word shows a consistent improvement on all testing scripts against both "No Co-training" and "2x Batch size", indicating a significant advantage over simply increasing the batch size.

This set of experiments directly answers RQ2: Though co-training with a single synthetic co-training task helps, it's better to co-train with both character and word-level synthetic data.

4.3 Generalized Zero Shot Learning Capability

We first benchmark the three configurations (the CJK only "Ours-CJK", the Indic only "Ours-IND", and the unified configuration "Ours-Uni") on the generalized zero-shot learning setups and compare them to existing Open-Set text recognition methods as applicable. Quantitative results are shown in Table 3, and qualitative results are shown in Fig. 7. Quantitative results suggest that the proposed framework achieves the second best performance on recognizing capabilities of unseen characters, only lagging behind WnA [21]. Note that WnA is an MoE-based method with flexible input size, while during testing, this method has a single monolithic input size of 32 × 128, which limits its performance.

Fig. 7: Qualitative results on common failure patterns of GZSL.

Table 3: Generalized Zero-shot learning performance. Results are given as word accuracy (ACR↑). Bold and Italic indicate the best and the second best performance.

	JPN	KR	Yi	BEN-Val	GUJ-Val	BEN-Test	GUJ-Test
OpenCCD [26]	41.31	-	-	-	-	-	-
SAVR [26]	42.58	-	-	-	-	-	-
CFOR [23]	44.47	22.14	-	-	-	-	-
WnA [21]	**48.02**	**24.00**	-	-	-	-	-
No-Cotrain-CJK	42.53	18.62	9.54	-	-	-	-
No-Cotrain-IND	-	-	-	4.80	6.28	5.12	5.76
Ours-CJK	45.92	*23.32*	**16.10**	-	-	-	-
Ours-IND	-	-	-	8.04	7.52	6.82	8.68
Ours-Uni	*46.30*	22.84	*14.22*	**12.88**	**9.72**	**9.02**	**10.04**

In addition, we took 50 images out of the synthetic Yi scripts and benchmarked the OpenAI ChatGPT-5.2-20251211 on it. Specifically, we use the following prompt alongside the image: "Do OCR and return only transcription (modern Yi). `\n` Charset: [...]" (the 1,165 characters in the list omitted in text, more details can be found in our GitHub repo). The GPT model yields an ACR of 0 and a CER of 148.44%. Note that it is an average of per-sample CER, which can be $\geq$ 100% if predictions are longer than GT. On this subset, the full framework, Ours-Uni, yields a CER of 48.67% and an ACR of 18.00%. This indicates a fundamental capability advantage in recognizing unseen characters against SoTA VLMs.

Qualitatively, we noticed the model confusing characters with only detail differences, suggesting that the model's perception of "to be" is not perfectly aligned to human perception. We also see sampler alignment issues where characters are split, repeated, merged, or go missing, especially severe with long samples, and cause catastrophic failures. This indicates that the robustness of the CAM-CTC structure is still not robust enough. Noticeably, it rejects characters if the input image is blurry or too stylized, even if the character categories do have templates registered, which is actually more desirable than hallucination and thus demands a future metric to quantitatively measure this behavior in a future protocol, together with annotation revisions that fix errors we found.

4.4 Open-Set Text Recognition Capability

In this set of experiments, we evaluate the ability to find out-of-set characters by testing the proposed framework on the OSTR setup. Results are shown in Table 4. The results suggest the model demonstrates a comparable recognition performance on seen characters, while having a slightly lower recall compared to recent methods like CFOR [23] and WnA [21]. Our proposed framework (Ours-CJK and Ours-Uni) is slightly more prone to hallucination compared to the SoTA models, it is still capable of rejecting 3 out of four samples that include out-of-set characters. The precision is roughly the same as SoTA. This means

Table 4: Performance on the Open-Set Text Recognition task [24]. **ACR** (Word Accuracy, ↑) and **FM** (F-measure for rejection, ↑) are the main metrics. Underline indicates unseen characters during training. **Bold** and *Italic* indicate the best and the second best performance, and asterisk mean vertical samples are used in training.

Registered	Out-of-Set	Name	**ACR**	Recall	Precision	**FM**
Shared Kanji,	Kana	OSOCR-Large [24]	58.57	24.46	93.78	38.80
Unique Kanji	Latin	OpenSAVR-XL [26]	72.33	72.96	92.62	81.62
		MOoSE-XL* [22]	64.80	80.49	89.12	84.50
		CFOR [23]	71.80	**86.36**	89.52	**87.91**
		WnA-XL*	**77.35**	*80.68*	*94.89*	*87.21*
		Ours-CJK	*75.43*	75.38	**95.30**	84.18
		Ours-Uni	75.38	78.34	94.59	85.70

when the model rejects a sample, it almost always includes one or more out-of-set characters.

5 Discussion

5.1 Conflicts Among Indic Scripts

In this work, we find that training all Indic scripts as one single task yields a severe negative impact on the overall performance, shown in Table 5. We report the average ACR over 100k, 120k, and 140k-th iterations from the first run, which shows a very significant pattern. However, we find that treating Punjabi and Tamil as independent script groups (as independent tasks) can significantly alleviate this issue. However, although polarized, training everything together shows significantly better performance on four of the scripts (Hindi, Malayalam, Kannada, and Marathi). The mechanism behind this remains to be explored together with better MoE-based approaches [21, 46] to alleviate this issue in a data-driven way.

5.2 Limitations

Although the model performs very well on CJK-related scripts and outperforms ChatGPT on the Yi script, its performance on the Yi script and unseen Indic scripts is still not feasible for zero-shot real-world applications.

Also, the current framework is only verified on scene text and fails to cover the handwritten modality used in most historical documents. Also, the experiments in this work fail to cover several script families, e.g., Arabic, etc.

Table 5: Inter-script conflict within the Indic script family, results are in ACR (↑).

Name	*BEN*	*GUJ*	**TAM**	**PUN**	HIN	MAL	KAN	MAR	TEL	ORI
All in one (val)	0.04	0.00	0.12	0.52	**86.35**	**78.89**	**82.00**	**88.91**	0.01	0.00
Ours-IND (val)	**4.73**	**6.43**	**77.59**	**82.37**	82.77	69.67	76.41	86.24	**66.64**	**71.81**

5.3 Conclusion

Besides proposing a unified open-set text recognition framework that is capable of recognizing multiple unseen scripts across several script families, this work answers the research questions posed in Section 1:

RQ1. Can we scale open-set text recognition beyond CJK? Yes, the proposed framework (Ours-Uni) managed to achieve a word accuracy (ACR) of 14.22%, 9.02%, and 10.04for Yi, Bengali, and Gujarati respectively. This serves as a solid starting point as a unified open-set text recognition framework considering the vast character set sizes (around 1,000 for each script).

RQ2. How can we improve performance through co-training? The most feasible approach so far is co-training with both synthetic character and word-level data heterogeneously.

RQ3. Does human perception of "to be" generalize across scripts? Yes, but with nuance. On one hand, we notice that co-training with non-Indic scripts significantly boosts performance on Indic scripts (Ours-Uni vs. Ours-IND), indicating that certain descriptive features are shared across scripts. On the other hand, we noticed the performance slightly drops on the Korean and Yi benchmarks when co-trained with the Indic family (Ours-Uni vs. Ours-CJK). Also, training all Indic scripts together yields performance collapse for some scripts (Table 5). These two pieces of evidence indicate that the definition of "to be" is not always trivially generalizable.

Summary In summary, this proposed method serves as a solid baseline for a unified, script-agnostic open-set text recognition framework. The reason for the inter-script conflicts and its data-driven alleviation form our next research topic.

Acknowledgment

This work is supported by: **WASP**: the Wallenberg AI, Autonomous Systems and Software Program, with co-funding from Luleå Technical University;
WireTEN-C: a research and innovation project led by Luleå University of Technology together with regional and national actors.
The computation of this work is supported by: **Alvis**: a resource of the National Academic Infrastructure for Super computing in Sweden (NAISS); **Berzelius**: a resource of the National Supercomputer Center (NSC).

References

1. Ao, X., Li, X.H., Zhang, X.Y., Liu, C.L.: Prototype calibration with synthesized samples for zero-shot Chinese character recognition. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6295–6299. IEEE (2024)
2. Baek, J., Matsui, Y., Aizawa, K.: Cross-lingual learning in multilingual scene text recognition. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6770–6774. IEEE (2024)
3. Borisyuk, F., Gordo, A., Sivakumar, V.: Rosetta: Large scale system for text detection and recognition in images. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, KDD 2018, London, UK, August 19-23, 2018. pp. 71–79. ACM (2018)
4. Cao, Z., Lu, J., Cui, S., Zhang, C.: Zero-shot handwritten Chinese character recognition with hierarchical decomposition embedding. Pattern Recognit. **107**, 107488 (2020)
5. Caruana, R.: Multitask learning. Machine Learning **28**(1), 41–75 (1997)
6. Chanda, S., Baas, J., Haitink, D., Hamel, S., Stutzmann, D., Schomaker, L.: Zero-shot learning based approach for medieval word recognition using deep-learned features. In: 16th International Conference on Frontiers in Handwriting Recognition, ICFHR 2018, Niagara Falls, NY, USA, August 5-8, 2018. pp. 345–350. IEEE Computer Society (2018)
7. Chanda, S., Haitink, D., Prasad, P.K., Baas, J., Pal, U., Schomaker, L.: Recognizing Bengali word images - A zero-shot learning perspective. In: 25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021. pp. 5603–5610. IEEE (2020)
8. Chang, W., You, T., Seo, S., Kwak, S., Han, B.: Domain-specific batch normalization for unsupervised domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 7354–7362. Computer Vision Foundation / IEEE (2019)
9. Chen, J., Li, B., Xue, X.: Zero-shot Chinese character recognition with stroke-level decomposition. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021. pp. 615–621. ijcai.org (2021)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE conference on Computer Vision and Pattern Recognition. pp. 248–255. IEEE (2009)
11. Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y.: Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 7098–7107. Computer Vision Foundation / IEEE (2021)
12. Geng, C., Huang, S., Chen, S.: Recent advances in open set recognition: A survey. IEEE Trans. Pattern Anal. Mach. Intell. **43**(10), 3614–3631 (2021)
13. Gong, L., Xiong, C., Liu, X., Bajaj, P., Xie, Y., Cheung, A., Gao, J., Song, X.: Model-generated pretraining signals improves zero-shot generalization of text-to-text transformers. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 12933–12950 (2023)
14. Graves, A., Fernández, S., Gomez, F.J., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Machine Learning, Proceedings of the Twenty-Third International Conference

(ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006. ACM International Conference Proceeding Series, vol. 148, pp. 369–376. ACM (2006)
15. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 2315–2324. IEEE Computer Society (2016)
16. He, S., Schomaker, L.: Open set Chinese character recognition using multi-typed attributes (2018), http://arxiv.org/abs/1808.08993
17. Huang, J., Liang, K.J., Kovvuri, R., Hassner, T.: Task grouping for multilingual text recognition. In: European Conference on Computer Vision. pp. 297–313. Springer (2022)
18. Huang, Y., Jin, L., Peng, D.: Zero-shot Chinese text recognition via matching class embedding. In: International Conference on Document Analysis and Recognition. pp. 127–141. Springer (2021)
19. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. In: NIPS Deep Learning Workshop. Neural Information Processing Systems (2014)
20. Lin, Z., Liu, C., Zhang, R., Gao, P., Qiu, L., Xiao, H., Qiu, H., Lin, C., Shao, W., Chen, K., et al.: Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. arXiv preprint arXiv:2311.07575 (2023)
21. Liu, C., Barney Smith, E.H.: Watch and act: Multi-orientation open-set scene text recognition via dynamic expert routing. In: International Conference on Document Analysis and Recognition. pp. 595–612. Springer (2025)
22. Liu, C., Corbillé, S., Barney Smith, E.H.: MOoSE: Multi-orientation sharing experts for open-set scene text recognition. In: International Conference on Document Analysis and Recognition. pp. 93–110. Springer (2024)
23. Liu, C., Yang, C., Fang, Z., Qin, H.B., Yin, X.C.: CFOR: Character-first open-set text recognition via context-free learning. IEEE Transactions on Image Processing (2024)
24. Liu, C., Yang, C., Qin, H., Zhu, X., Liu, C., Yin, X.: Towards open-set text recognition via label-to-prototype learning. Pattern Recognit. **134**, 109109 (2023)
25. Liu, C., Yang, C., Yin, X.: Open-set text recognition via character-context decoupling. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 4513–4522. IEEE (2022)
26. Liu, C., Yang, C., Yin, X.C.: Open-set text recognition via shape-awareness visual reconstruction. In: International Conference on Document Analysis and Recognition. pp. 89–105. Springer (2023)
27. Liu, X., Huang, W., Chen, J., Wang, X., Peng, J.: OracleProtoPNet: Oracle character recognition with interpretability. In: International Conference on Document Analysis and Recognition. pp. 208–223. Springer (2025)
28. Lunia, H., Mondal, A., Jawahar, C.: Indicstr12: a dataset for Indic scene text recognition. In: International Conference on Document Analysis and Recognition. pp. 233–250. Springer (2023)
29. Mishra, A., Alahari, K., Jawahar, C.V.: Scene text recognition using higher order language priors. In: British Machine Vision Conference, BMVC 2012, Surrey, UK, September 3-7, 2012. pp. 1–11. BMVA Press (2012)
30. Mondal, A., Tulsyan, K., Jawahar, C.: Indic scene text on the roadside. In: International Conference on Document Analysis and Recognition. pp. 263–278. Springer (2024)

31. Ouyang, L., Qu, Y., Zhou, H., Zhu, J., Zhang, R., Lin, Q., Wang, B., Zhao, Z., Jiang, M., Zhao, X., et al.: OmniDocBench: Benchmarking diverse PDF document parsing with comprehensive annotations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24838–24848 (2025)
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (2021)
33. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020)
34. Rai, A., Krishnan, N.C., Chanda, S.: Pho(sc)net: An approach towards zero-shot word image recognition in historical documents. In: International conference on document analysis and recognition. pp. 19–33. Springer (2021)
35. Vandenhende, S., Georgoulis, S., Gansbeke, W.V., Proesmans, M., Dai, D., Gool, L.V.: Multi-task learning for dense prediction tasks: A survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(7), 3614–3633 (2022)
36. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
37. Wang, T., Xie, Z., Li, Z., Jin, L., Chen, X.: Radical aggregation network for few-shot offline handwritten Chinese character recognition. Pattern Recognit. Lett. **125**, 821–827 (2019)
38. Wei, H., Sun, Y., Li, Y.: Deepseek-ocr 2: Visual causal flow. arXiv preprint arXiv:2601.20552 (2026)
39. Xiao, B., Wu, H., Xu, W., Dai, X., Hu, H., Lu, Y., Zeng, M., Liu, C., Yuan, L.: Florence-2: Advancing a unified representation for a variety of vision tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4818–4829 (2024)
40. Xu, J., Guo, Z., Hu, H., Chu, Y., Wang, X., He, J., Wang, Y., Shi, X., He, T., Zhu, X., et al.: Qwen3-omni technical report. arXiv preprint arXiv:2509.17765 (2025)
41. Yang, J., Zhou, K., Li, Y., Liu, Z.: Generalized out-of-distribution detection: A survey. International Journal of Computer Vision **132**(12), 5635–5662 (2024)
42. Yu, H., Wang, X., Li, B., Xue, X.: Chinese text recognition with a pre-trained CLIP-like model through image-ids aligning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 11943–11952 (October 2023)
43. Zhang, C., Gupta, A., Zisserman, A.: Adaptive text recognition through visual matching. In: Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVI. Lecture Notes in Computer Science, vol. 12361, pp. 51–67. Springer (2020)
44. Zhang, Y., Shi, Y., Zhang, P., Zhao, Y., Yang, Z., Jin, L.: Megahan97k: A large-scale dataset for mega-category Chinese character recognition with over 97k categories. Pattern Recognition p. 111757 (2025)
45. Zhao, S., Quan, R., Zhu, L., Yang, Y.: Clip4str: A simple baseline for scene text recognition with pre-trained vision-language model. IEEE Transactions on Image Processing **33**, 6893–6904 (2024)

46. Zheng, T., Chen, Z., Huang, B., Zhang, W., Jiang, Y.G.: MRN: Multiplexed routing network for incremental multilingual text recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18644–18653 (2023)
47. Zu, X., Yu, H., Li, B., Xue, X.: Chinese character recognition with augmented character profile matching. In: MM '22: The 30th ACM International Conference on Multimedia, Lisbon, Portugal, October 10 - 14, 2022. pp. 6094–6102. ACM (2022)