BornoDrishti: Leveraging Vision Encoders and Domain-Adaptive Learning for Bangla OCR on Diverse Documents

S M Jishanul Islam[2], **Md Mehedi Hasan**[2], **Masbul Haider Ovi**[2]
AKM Shahariar Azad Rabby[2], **Fuad Rahman**[1]

[1]Apurba Technologies, CA, USA
[2]Apurba Technologies Ltd, Dhaka, Bangladesh
Correspondence: rabby@apurbatech.com

Abstract

OCR for Bangla scripts remains a challenging problem, with existing solutions limited to single-domain processing. Current approaches lack a unified vision encoder that can understand diverse Bangla script variations, hindering practical deployment. We present BornoDrishti, the first unified OCR system based on the vision transformer that accurately recognizes both printed and handwritten Bangla scripts within a single model. Our approach introduces a novel domain objective that enables the model to learn domain-invariant representations while preserving script-specific features, eliminating the need for separate domain experts. BornoDrishti achieves competitive accuracy across both domains, setting state-of-the-art performance for printed scripts and demonstrating that a single unified model can match or exceed specialized uni-domain systems. We evaluate our model against state-of-the-art domain-specific and cross-domain OCR systems. This work establishes a foundation for advancing practical applications by using a unified multi-domain OCR system for complex Bangla scripts.

1 Introduction

Optical character recognition (OCR) is one of the domains of computer vision that has seen significant advances following the introduction of vision transformers (ViT). In recent times, OCR models, including GOT OCR (Wei et al., 2024) and the Qwen VL series (Bai et al., 2025), have been using vision transformers as backbone architectures. These multilingual models are also showing notable results in low-resource languages (e.g., Bangla). However, these models suffer from domain adaptation due to the bias in their training data towards computer-composed documents.

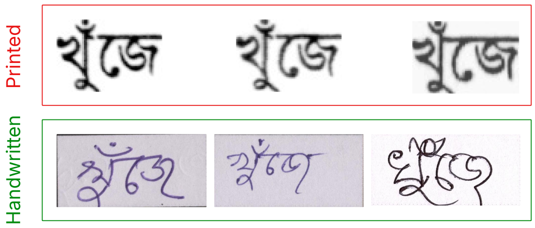

Figure 1: The variation in stroke patterns, writing styles, and character shapes in printed and handwritten Bangla documents

This challenge gets more difficult when applying these models for Bangla and other Indic languages. Bangla language exhibits extensive intra-domain diversity: variations arising from differences in individual handwriting styles, stroke patterns, character shapes, and writing fluency (Figure 1). Generalizing across intra-domain diversity in handwritten text is a highly challenging task for any OCR model. Beyond this, the problem becomes even more formidable when inter-domain diversity is taken into account. Printed documents preserve uniform spacing, generic stroke patterns, and character shapes. Together, these intra- and inter-domain diversities pose complex challenges for developing a robust, generalized OCR solution for Bangla.

To solve these problems, we create BornoDrishti, a single vision encoder that recognizes Bangla words in any script style, whether printed or handwritten. To the best of our knowledge, this is the first encoder model trained in a contrastive manner to adapt to different domains for Bangla OCR. We start by taking the vision transformer (Dosovitskiy et al., 2021) and training it CLIP-style to learn the various styles for each word. To enable it to understand both printed and handwritten scripts, we use a progressive learning approach in three stages. During cross-domain training, we incorporate domain information into the loss function to define a novel domain objective. This enables the model to understand the image's domain and its OCR label.

BornoDrishti sets a new paradigm for advancing Bangla OCR, moving beyond single-expert models

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track), pages 278–286
March 25-27, 2026 ©2026 Association for Computational Linguistics

to create more generalized encoders. Our main contributions include: the first contrastive-style Bangla OCR encoder to work across multiple domains, a domain objective to optimize the model based on the domain alongside the output, and a practical encoder model that can replace existing domain-specific encoders, thus saving compute resources. The Government of Bangladesh will soon open-source the code. The end-to-end OCR demo is available at http://kagoj.ai/.

2 Related Work

Bangla OCR has been enhanced by Apurba Technologies Ltd over the years. Their initial works were focused on character-level OCR models. In 2021, they introduced two CNN models for printed and handwritten documents. One model used a chained head output module, and another used a multi-headed CNN (Rabby et al., 2021; Islam et al., 2021). Both models achieved a CRR score of over 95% across all document types. Later in 2022, they presented a character-level solution based on resunet++ for low-resource languages such as Bangla and Assamese (Das et al., 2022). This work was followed by a knowledge distillation method with CRNN-based models (Hossain et al., 2022). They used a shallow CNN and the ResNet18 model as the teacher model, and a VGG-based CRNN with a BiLSTM layer as the student model, achieving 74.40% and 84.46% CRR score on BN-HTRd and BanglaWriting datasets (Rahman et al., 2023; Mridha et al., 2021). Later in 2023, they presented another OCR system with specialized segmentation models (Rabby et al., 2024). They introduced a self-attention VGG-based multi-headed neural network architecture for OCR that was capable of understanding various document types, including computer-composed, typewriter, letterpress, and handwritten documents. It achieved an average accuracy of 87.20% on the Levenshtein distance-based metric and 98.05% accuracy on the Confusion matrix-based metric across all document types.

Apart from these significant works, a few works are presented by individuals. A transformer model was proposed in 2023 that used a ViT-based architecture as the image encoder and RoBERTa as the text decoder (Hasan et al., 2024). This model achieved a CER score of 0.07 and A WER score of 0.12 on Bangla text. APSIS-Net was introduced in 2024 (Zulkarnain et al., 2023). This word recognition model comprises a CNN-based attention encoder for images and a positional embedding layer, achieving 0.59 CER and 0.80 WER on word-level image recognition. Another work was introduced in 2024 that focused on Bangla handwritten character recognition, leveraging an ensemble learning technique (Haque et al., 2024). They used ResNet and Google LeNet to achieve 98.00% accuracy on Bangla handwritten characters.

Our approach builds on established techniques from the domain adaptation literature. Domain-Adversarial Neural Networks (DANN) introduced gradient reversal to learn domain-invariant features by training a domain discriminator adversarially against the feature extractor (Ganin et al., 2016). This principle has been extended in works such as Adversarial Discriminative Domain Adaptation (ADDA), which decouples source and target encoders during adaptation (Tzeng et al., 2017), and Deep CORAL, which aligns second-order statistics across domains (Sun and Saenko, 2016). Deep Adaptation Networks (DAN) minimize distributional discrepancy via the multi-kernel maximum mean discrepancy (Long et al., 2015). Progressive training strategies have also been explored for domain adaptation. Curriculum learning establishes that ordering training samples from easy to hard improves convergence and generalization (Bengio et al., 2009). This principle has been applied to domain adaptation through Progressive Feature Alignment Networks (PFAN), which gradually align features across domains (Chen et al., 2019), and self-paced learning approaches that adaptively weight samples during training (Kumar et al., 2010).

Existing works have significant gaps that need to be addressed. Character-level approaches require an additional character-level segmentation model. Some of the works suffer from punctuation restoration, which is essential for preserving the semantic meaning in Bangla text. Other transformer and ensemble models do not address multi-type documents. In addition, while progressive learning has achieved success in general computer vision tasks, its application to low-resource script OCR with significant intra-domain variability remains underexplored. Our work combines CLIP-style contrastive learning with gradient reversal and progressive training, specifically tailored for the unique challenges of cross-domain Bengali script recognition—where printed and handwritten text exhibit fundamentally different visual characteristics yet share the same character vocabulary.

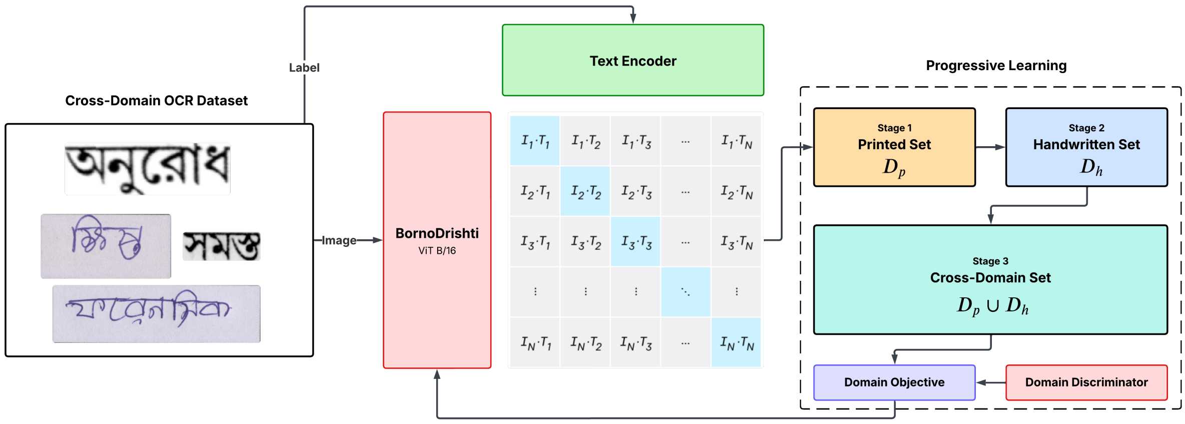
Label
Text Encoder
Progressive Learning
Cross-Domain OCR Dataset
অনুরোধ
সমস্ত
Image
BornoDrishti
ViT B/16
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
Stage 1
Printed Set
D_p
Stage 2
Handwritten Set
D_h
Stage 3
Cross-Domain Set
$D_p \cup D_h$
Domain Objective
Domain Discriminator

Figure 2: BornoDrishti is a single vision encoder trained to recognize cross-domain Bangla scripts using CLIP and a domain objective

3 BornoDrishti

3.1 Problem Formulation

Bengali OCR remains challenging to deploy in real production environments because the script contains more than 50 basic characters, over 400 conjunct forms, and significant visual variability across printed and handwritten sources. In real-life documents, such as bank forms, government records, ID documents, and handwritten applications, these two domains frequently appear together, often within the same page. As a result, OCR systems must reliably handle inconsistent spacing, irregular stroke patterns, stylistic differences, and noisy scan quality.

Given an input word image $I \in \mathbb{R}^{(H \times W \times C)}$, the OCR goal is to output a sequence of Bengali characters $Y = y_1, y_2, ..., y_n$ where each y_i belongs to a predefined character vocabulary V. The prediction model aims to estimate:

$$P(Y|I;\theta) = \prod_{i}^{Y} P(y_i|y_1, ..., y_{i-1}, I;\theta) \quad (1)$$

In practice, however, a single function $f_\theta : I \rightarrow Y$ rarely generalizes well across all document types. Current Bengali OCR systems deployed in industry address this by maintaining separate models for printed and handwritten text:

$$\begin{aligned} f_{printed} &= I_p \rightarrow Y \quad \text{for printed text} \\ f_{handwritten} &= I_h \rightarrow Y \quad \text{for handwritten text} \end{aligned}$$

Here, $I_p \in D_{printed}$ and $I_h \in D_{handwritten}$ belong to visually distinct domains with a clear distribution shift $P(I_p) \neq P(I_h)$. Maintaining and serving multiple domain-specific OCR models increases operational overhead including domain-routed deployments, additional compute resources, and manual configuration to handle mixed document types.

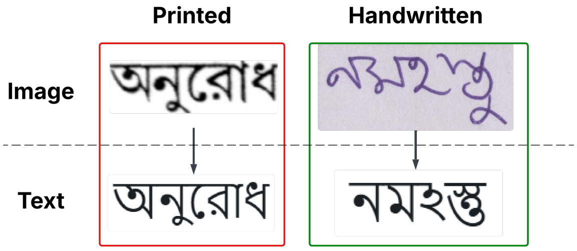

Figure 3: Example of the printed and handwritten sample in the dataset

To address this practical bottleneck, we leverage a CLIP-based vision encoder that is naturally exposed to diverse visual patterns. Our objective is to adapt this encoder to capture both printed and handwritten Bengali script styles within a single unified model, reducing system complexity while improving robustness across real-world document variations.

3.2 Dataset

The studies in recent years have resulted in a few datasets for Bangla OCR, including CMATERdb (Sarkar et al., 2012), Banglalekha-isolated (Biswas et al., 2017), Ekush (Rabby et al., 2019), Bengali.AI dataset (Alam et al., 2020), BanglaWriting (Mridha et al., 2021), BN-HTRd (Rahman et al., 2023), and IIIT-INDIC-HW-WORDS-Bengali (Gongidi and Jawahar, 2021). However, most of these datasets are either character-level datasets or document-level, which require word synthesis. Only IIIT-INDIC-HW-WORDS-Bengali is a word-level handwritten dataset that aligns with our task. It contains images of 113K words (11,295 unique words), written by 24 people from diverse educational backgrounds and age groups, resulting in a notable diversity in the writing patterns. To address the cross-domain target, we merged the Mozhi-Bengali dataset with the handwritten dataset. The Mozhi-Bengali dataset contains 100K word images (18,352 unique words) collected from 1,000 printed document pages. Figure 3 shows the examples in the dataset.

3.3 BornoDrishti

We create BornoDrishti, the first self-supervised language-image alignment method for domain adaptation in Bangla OCR. We take a vision backbone and train it across multiple stages with a domain objective, equipping it to capture cross-domain scripts for Bangla. The entire flow is shown in Figure 2.

3.3.1 Initial Architecture

Our goal was to build a single OCR encoder that works reliably on both printed and handwritten Bengali text without maintaining separate domain-specific models. For this, we use a ViT-B/16 (Dosovitskiy et al., 2021) backbone paired with a CLIP-style contrastive learning setup (Radford et al., 2021). This choice is motivated by three practical considerations: ViT models are stable across diverse visual patterns, CLIP pretraining provides strong initialization for low-resource scripts, and the architecture runs efficiently in production on a single GPU.

Image Encoder. The ViT encoder takes a word-level image, extracts visual patches, and produces a single embedding vector through the `[CLS]` token. Instead of training the encoder from scratch, we fine-tune it with contrastive supervision using ground-truth text labels. This allows the encoder to learn character-level structure without requiring an explicit decoder during training. The resulting image embedding is compact and suitable for large-scale batch inference.

Text encoder. For text, we use a lightweight GPT-2 model (Radford et al., 2019) to compute a dense representation of each target word. Although the final system does not rely on natural-language generation, using a text encoder provides a stable semantic space for contrastive alignment. The output from the `[EOS]` position is projected into the same embedding dimension as the image encoder.

Contrastive alignment. Given a batch of image–text pairs, we compute similarity scores using a temperature-scaled dot product and optimize a symmetric contrastive loss. This setup encourages the model to pull together matched image–text pairs while pushing apart mismatched pairs. In practice, this objective is more straightforward to optimize than a full autoregressive OCR decoder and yields representations that generalize well across both printed and handwritten styles.

Design rationale. This architecture minimizes deployment complexity and training instability. It avoids heavy decoders, reduces the number of model components that must be maintained, and supports faster inference. Most importantly, the contrastive formulation provides a unified representation space in which both printed and handwritten word images can be jointly learned, forming the foundation of our cross-domain OCR system.

3.3.2 Progressive Learning

In practice, training a single OCR model on a mix of printed and handwritten Bengali data leads to unstable convergence. Early experiments showed that the encoder overfits to printed samples because they are visually consistent, while handwritten samples introduce high variability in stroke width, curvature, spacing, and writing style. Training on both domains from the start leads to frequent oscillations in loss and poor generalization on handwritten text.

To address this, we follow a progressive learning strategy that stabilizes the encoder before exposing it to full domain diversity. The training process is divided into three stages: printed-only, handwritten-only, and mixed-domain. At first, the encoder is trained on printed word images. This allows the model to learn clean structural patterns and basic character shapes. Next, the model is then fine-tuned on handwritten samples. The initialization from stage one helps the encoder adapt to higher variability without collapsing. Finally, the model is trained on both domains together. At this stage, the encoder learns to align printed and handwritten representations within a shared embedding space. We select the best checkpoint from each stage based on validation performance and use the final mixed-domain checkpoint for all deployments. This staged process improves stability and leads to significantly better cross-domain accuracy,

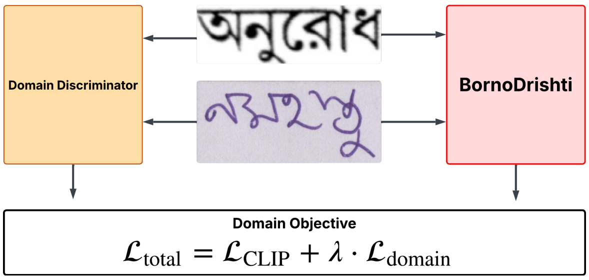
Domain Discriminator
অনুরোধ
BornoDrishti
Domain Objective
$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CLIP}} + \lambda \cdot \mathcal{L}_{\text{domain}}$

Figure 4: The flow of the domain objective. The Domain Discriminator predicts the current domain and passes the information to the shared loss

especially on challenging handwritten inputs.

3.3.3 Domain Objective

Even with progressive learning, handwritten samples remain more complex to model due to their greater visual variation. To help the encoder learn features that transfer across domains, we introduce an auxiliary domain objective during the final training stage. We attach a small MLP-based discriminator to the image embedding and train it to predict whether the input came from the printed or handwritten domain. During backpropagation, we apply gradient reversal to encourage the encoder to remove domain-specific cues and learn representations that generalize better.

Let $y_i \in \{0, 1\}$ denote the domain label (0 for printed, 1 for handwritten) and g_ϕ be the discriminator. The output by the discriminator is denoted by O_{g_ϕ}. The discriminator is trained with binary cross-entropy:

$$\begin{aligned}\mathcal{L}_{\text{domain}} = -\mathbb{E}\big[& y_i \log g_\phi(z_i) \\ & + (1 - y_i) \log (1 - g_\phi(z_i))\big]\end{aligned} \tag{2}$$

where z_i is the image embedding from the vision encoder. The total loss becomes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CLIP}} + \lambda \cdot \mathcal{L}_{\text{domain}} \tag{3}$$

This formulation implements a min-max adversarial objective. The discriminator g_ϕ minimizes $\mathcal{L}_{\text{domain}}$ to correctly classify domains, while the encoder, through the gradient reversal layer, effectively maximizes this term by learning representations that confuse the discriminator. The reversed gradients encourage the encoder to suppress domain-specific visual cues (e.g., uniform stroke width in printed text vs variable strokes in handwriting) and instead capture domain-invariant character features.

We set $\lambda = 0.1$ based on validation performance, balancing the contrastive alignment objective with domain invariance. To prevent the domain loss from destabilizing training, we apply gradient clipping with a maximum norm of 1.0. The domain objective is enabled only in the final mixed-domain stage; earlier stages benefit from learning domain-specific features before encouraging invariance. In the final stage, the domain objective helps align both domains into a common embedding space without harming printed performance. This approach improves the model's ability to recognize handwritten words. It reduces the performance gap between domains, which is essential for deployment in real OCR workflows that process mixed-type documents.

4 Experiments

4.1 Experimental Setup

We implement BornoDrishti using PyTorch and HuggingFace. BornoDrishti was trained for 50 epochs using our progressive learning strategy. The experiments and ablations were also conducted for the same number of epochs. The batch size is set to 64 to accommodate the GPU memory. The optimizer used was Adam (Kinga et al., 2015), and different instances were set for each training stage with learning rates set to $5e^{-5}$, $2e^{-5}$, and $1e^{-5}$, respectively, with cosine LR scheduling to adjust it during training. All experiments were conducted on a 1×T4 GPU with 16GB VRAM to ensure suitability for resource-constrained environments.

4.2 Evaluation Metrics

We evaluate BornoDrishti using two comprehensive metrics: Top-1 and Top-5 Accuracy. The Top-1 accuracy specifies the exact match accuracy of the word predicted, and the Top-5 accuracy specifies the chance that the correct label appears among the top 5 predictions made by the model. These assess both classification accuracy and cross-modal retrieval capabilities, reflecting the CLIP-style training approach. These retrieval metrics validate that our model learns meaningful cross-modal representations rather than merely memorizing image-text pairs. Since related benchmarks commonly report the "accuracy" metric, we use Top-1 Accuracy as the standard for easier comparison of results. BornoDrishti is an encoder-only cross-domain model trained using CLIP-style objectives. Unlike sequence-decoders, CLIP-based

Model	Overall Acc. (%)	Cross-Domain
Tesseract OCR (Smith, 2007)	57.56	✓
Bengali OCR (Rabby et al., 2024)	87.20	×
BornoDrishti (w/o progressive learning)	**77.76**	✓
BornoDrishti (w progressive learning)	**83.77**	✓

Table 1: The performative comparison between our proposed model and other state-of-the-art methods.

encoders naturally produce a cross-modal embedding space. For such encoders, the standard OCR metrics (CER/WER) do not directly apply because they require a decoder architecture. Thus, we use Top-1 and Top-5 retrieval accuracies, which is the canonical metric for alignment-based encoders.

4.3 Results

The overall results demonstrate that BornoDrishti performs on par with state-of-the-art encoder-based Bangla OCR models. As shown in Table 1, Bengali OCR (Rabby et al., 2024) exceeds the proposed model's performance by 4%. However, it is not domain adaptive. It requires a separate model for each domain, adding computational cost during initialization and deployment. This results in their model requiring 2×T4s to serve up OCR outputs, in contrast to a single T4 used by BornoDrishti. Compared to Google's Tesseract OCR (Smith, 2007), an industry standard in OCR, our model achieves significantly higher overall accuracy, highlighting the stark gap in Bangla OCR across industry-grade systems. Tesseract is domain-adaptive, but cannot capture the cross-domain script styles for Bangla. In terms of computational resources, Tesseract does not require a GPU to run and is primarily optimized for CPU-based inference. This is a stark contrast in optimization to BornoDrishti. However, we trade off the CPU optimization for low-compute unified cross-domain inference. Recent OCR systems such as Donut, TrOCR, GOT-OCR, and Qwen-VL are full sequence-to-sequence VLM architectures that integrate both an encoder and a decoder, making them fundamentally different from our encoder-only formulation. Therefore, we restrict comparisons to encoder-level or domain-specific OCR systems aligned with our scope.

4.4 Ablations

We conduct a few ablations to back our methodology. These include verifying the effectiveness of both the progressive learning training recipe and performance across uni- and cross-domain settings.

Method	Top-1	Top-5
Comb. w/o prog. learning	77.7%	94.7%
Comb. w prog. learning	83.7%	96.8%

Table 2: The performance of the Top-1 and Top-5 accuracy scores for training without and with progressive learning, recorded in percentages. "Comb." abbreviates to "Combined", "w/o" to "without", "w" to "with", and "prog" to "progressive". Words have been shortened for space.

4.4.1 Does the training recipe affect model performance?

The metrics recorded for the training recipe are shown in Table 2. It is observed that if we train the domains within a combined set without progressive learning, the Top-1 accuracy is 77.7%. This is a significant drop in performance compared to the Top-1 accuracy of 83.7% achieved during progressive learning. We attribute this to the ability of progressive learning: specifically, the model first understands the patterns through printed examples, then builds on that knowledge to adapt to handwritten examples, and finally learns both at once.

4.4.2 How does the model perform in single domain settings?

The metrics recorded for performance across domains are shown in Table 3. For our model, we report the Top-1 accuracy. The uni-domain accuracies were recorded during testing after training the model separately for that specific domain. Compared to (Rabby et al., 2024), it can be observed that the performance of our model on the printed domain is high, while the performance of the handwritten domain is significantly low. However, (Rabby et al., 2024) uses domain-specific CNNs, with no cross-domain models available. Thus, we record that the model has no capability across domains. Compared to Tesseract (Smith, 2007), our model outperforms it across all domains, revealing a significant gap in its OCR capabilities for Bangla scripts and demonstrating strong cross-domain performance. In addition to Top-1

Domain (Work)	Accuracy (%)
Pr-only (Smith, 2007)	73.49
Hwr-only (Smith, 2007)	35.80
Pr+Hwr-only (Smith, 2007)	44.64
Pr-only (Rabby et al., 2024)	90.06
Hwr-only (Rabby et al., 2024)	86.84
Pr+Hwr (Rabby et al., 2024)	NC
Pr-only (Ours)	94.7
Hwr-only (Ours)	68.9
Pr+Hwr (Ours)	83.77

Table 3: The comparisons between the performances when applied to uni-domains with combined domains. The "Pr." is abbreviated to "Printed", "Hwr" to "Handwritten", and "NC" to "Not capable".

accuracy, we report Top-5 accuracy in uni- and cross-domain settings for our retrieval-based model. The Top-5 accuracies for printed, handwritten, and cross-domain are 99.74%, 95.22%, and 96.89%, respectively. These results show that BornoDrishti can retrieve accurate results.

5 Discussion

5.1 Handwritten Performance Gap

As shown in Tables 1 and 3, there is a great difference in the performance gaps in identifying handwritten images. The performance gap between printed and handwritten domains stems from fundamental differences in visual complexity. Handwritten Bengali exhibits: (1) high inter-writer variability in stroke patterns and character formation, (2) inconsistent spacing and baseline alignment, and (3) degraded image quality from scanning handwritten documents. The datasets include samples from only 24 writers, limiting exposure to the full distribution of handwriting styles.

To bridge this gap, we identify several promising directions: (1) augmenting handwritten training data with synthetic variations using elastic deformations and style transfer, (2) incorporating writer-adaptive layers that capture individual writing characteristics, and (3) leveraging self-training with pseudo-labels from high-confidence predictions on unlabeled handwritten data. These extensions represent our immediate future work toward achieving parity with domain-specific models such as (Rabby et al., 2024), while maintaining the deployment benefits of a unified architecture.

5.2 Deployment and Extensions

BornoDrishti is currently used in our product as a unified cross-domain vision encoder. Due to its lightweight design, with around 86 million parameters, it saves compute resources by running on a single T4 in an AWS EC2 instance. At inference time, BornoDrishti processes individual word images in approximately 15ms on a T4 GPU, which is 3-5ms slower than the CNN-based encoder of (Rabby et al., 2024). This marginal latency overhead is attributable to the ViT architecture's self-attention computation. However, for mixed-domain documents, BornoDrishti eliminates the need for domain classification and model switching, resulting in easier end-to-end processing compared to multi-model pipelines.

6 Conclusion

We present BornoDrishti, the first self-supervised language-image alignment method for domain adaptation in Bangla OCR, with a lightweight model architecture for on-production deployment. We demonstrate that, for cross-domain Bangla OCR, progressive learning is highly recommended for domain adaptation. Furthermore, we introduce the domain objective, which penalizes examples not only based on word prediction but also on the domain, forcing the model to learn the domain-specific script styles a word will exhibit. We compare our model to other industry-grade Bangla OCR systems and demonstrate significant improvements in accuracy and capabilities. We discuss the current usage of BornoDrishti in production pipelines and outline its next stage of improvement. We create BornoDrishti as one of our steps towards creating an end-to-end VLM-based document OCR model for Bangla.

Limitations

While the work shows promising directions in incorporating CLIP-trained encoders to Bangla OCR in resource-constrained production environments, the current job is limited to only two domains: printed and handwritten. There are many Bangla scripts, including letterpress and typewriter scripts. While an internal dataset of diverse images is being prepared, this work aims to make an initial observation on the use of such training methods in low-resource environments, a need in countries with limited computational resources.

References

Samiul Alam, Tahsin Reasat, Asif Shahriyar Sushmit, Sadi Mohammad Siddiquee, Fuad Rahman, Mahady Hasan, and Ahmed Imtiaz Humayun. 2020. Multilabel classification of common bengali handwritten graphemes: Dataset and challenge. *arXiv preprint arXiv: 2010.00170*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Mithun Biswas, Rafiqul Islam, Gautam Kumar Shom, Md. Shopon, Nabeel Mohammed, Sifat Momen, and Anowarul Abedin. 2017. Banglalekha-isolated: A multi-purpose comprehensive dataset of handwritten bangla isolated characters. *Data in Brief*, 12:103–107.

Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. 2019. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 627–636.

Avishek Das, AKM Shahariar Azad Rabby, Ibna Kowsar, and Fuad Rahman. 2022. A deep learning-based unified solution for character recognition. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1671–1677.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35.

Santhoshini Gongidi and CV Jawahar. 2021. iiit-indic-hw-words: A dataset for indic handwritten text recognition. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part IV 16*, pages 444–459. Springer.

Farhanul Haque, Md Al-Hasan, Sumaiya Tabssum Mou, Abu Saleh Musa Miah, Jungpil Shin, and Md Abdur Rahim. 2024. Multichannel attention networks with ensembled transfer learning to recognize bangla handwritten charecter. *arXiv preprint arXiv:2408.10955*.

SM Hasan, Aakar Dhakal, Md Humaion Kabir Mehedi, and Annajiat Alim Rasel. 2024. Optical text recognition in nepali and bengali: A transformer-based approach. *arXiv preprint arXiv:2404.02375*.

Md. Ismail Hossain, Mohammed Rakib, Sabbir Mollah, Fuad Rahman, and Nabeel Mohammed. 2022. Lila-boti : Leveraging isolated letter accumulations by ordering teacher insights for bangla handwriting recognition. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1770–1776.

Md. Majedul Islam, Avishek Das, Ibna Kowsar, A K M Shahariar Azad Rabby, Nazmul Hasan, and Fuad Rahman. 2021. Towards building a bangla text recognition solution with a multi-headed cnn architecture. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1061–1067.

Diederik Kinga, Jimmy Ba Adam, and 1 others. 2015. A method for stochastic optimization. In *International conference on learning representations (ICLR)*, volume 5. California;.

M Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR.

M.F. Mridha, Abu Quwsar Ohi, M. Ameer Ali, Mazedul Islam Emon, and Muhammad Mohsin Kabir. 2021. Banglawriting: A multi-purpose offline bangla handwriting dataset. *Data in Brief*, 34:106633.

AKM Shahariar Azad Rabby, Hasmot Ali, Md. Majedul Islam, Sheikh Abujar, and Fuad Rahman. 2024. Enhancement of bengali ocr by specialized models and advanced techniques for diverse document types. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 1102–1109.

AKM Shahariar Azad Rabby, Sadeka Haque, Md. Sanzidul Islam, Sheikh Abujar, and Syed Akhter Hossain. 2019. Ekush: A multipurpose and multitype comprehensive database for online off-line bangla handwritten characters. In *Recent Trends in Image Processing and Pattern Recognition*, pages 149–158, Singapore. Springer Singapore.

Akm Shahariar Azad Rabby, Md. Majedul Islam, Zahidul Islam, Nazmul Hasan, and Fuad Rahman. 2021. Towards building a robust large-scale bangla text recognition solution using a unique multiple-domain character-based document recognition approach. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1393–1399.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and

1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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*, pages 1–16. Chapman and Hall/CRC.

Ram Sarkar, Nibaran Das, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri, and Dipak Kumar Basu. 2012. Cmaterdb1: a database of unconstrained handwritten bangla and bangla–english mixed script document image. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(1):71–83.

R. Smith. 2007. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633.

Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, and 1 others. 2024. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. *arXiv preprint arXiv:2409.01704*.

Imam Mohammad Zulkarnain, Shayekh Bin Islam, Md Zami Al Zunaed Farabe, Md Mehedi Hasan Shawon, Jawaril Munshad Abedin, Beig Rajibul Hasan, Marsia Haque, Istiak Shihab, Syed Mobassir, MD Ansary, and 1 others. 2023. bbocr: An open-source multi-domain ocr pipeline for bengali documents. *arXiv preprint arXiv:2308.10647*.