Adapting Vision-Language Models for Hindi OCR

Shaon Bhattacharyya*[1][0009−0009−8452−9043], Souvik Ghosh*[1][0009−0004−3300−0266], Prantik Deb*[2][0009−0008−9585−5664], Ajoy Mondal[1][0000−0002−4808−8860], and C. V. Jawahar[1][0000−0001−6767−7057]

[1] CVIT, International Institute of Information Technology, Hyderabad, India
{shaon.b,souvik.g}@research.iiit.ac.in, {ajoy.mondal,jawahar}@iiit.ac.in
[2] BCCL, International Institute of Information Technology, Hyderabad, India
prantik.d@research.iiit.ac.in

Abstract. Optical Character Recognition (OCR) for Indian languages is challenging due to diverse scripts, complex characters, and varied writing styles. This work presents *HindiOCR-VLM*, a Vision-Language Model adapted for Hindi OCR using Low-Rank Adaptation (LoRA) on a model pre-trained for Chinese and English. We explore two adaptation strategies: (i) a *single-stage* model that directly predicts text at the page level — bypassing intermediate word or line detection — focused on printed documents, and (ii) a *two-stage* model for multi-domain scenarios (printed, handwritten, and scene text), which first detects words and then recognizes them individually. Inspired by human learning processes, we propose a progressive learning approach — a training strategy to the *single-stage* model to enhance language acquisition and accelerate convergence. Leveraging the vision encoder's rich representations, our method enables effective multi-domain training. Experimental results and ablation studies show that *HindiOCR-VLM* handles the complexities of the Devanagari script well and outperforms domain-specific models, offering a unified and robust solution for Hindi OCR. Code and resources are available at: https://hindiocr-vlm.github.io/.

Keywords: OCR · Indian script and language · Vision-Language Model · Low-Rank Adaptation · progressive learning · multi-domain.

1 Introduction

OCR has progressed from rule-based and statistical methods (e.g., HMMs [7,17, 46], SVMs [4]) to deep learning, with LSTMs [3,18,40,47] significantly improving sequential text recognition. Transformer-based models like TrOCR [34] enabled parallel processing and long-range attention. Recent advances involve Vision-Language Models (VLMs) [2, 10, 13, 52], which integrate visual and linguistic cues for deeper text understanding. VLMs surpass traditional OCR by leveraging language priors to interpret ambiguous or degraded text, significantly improving

* These authors contributed equally.

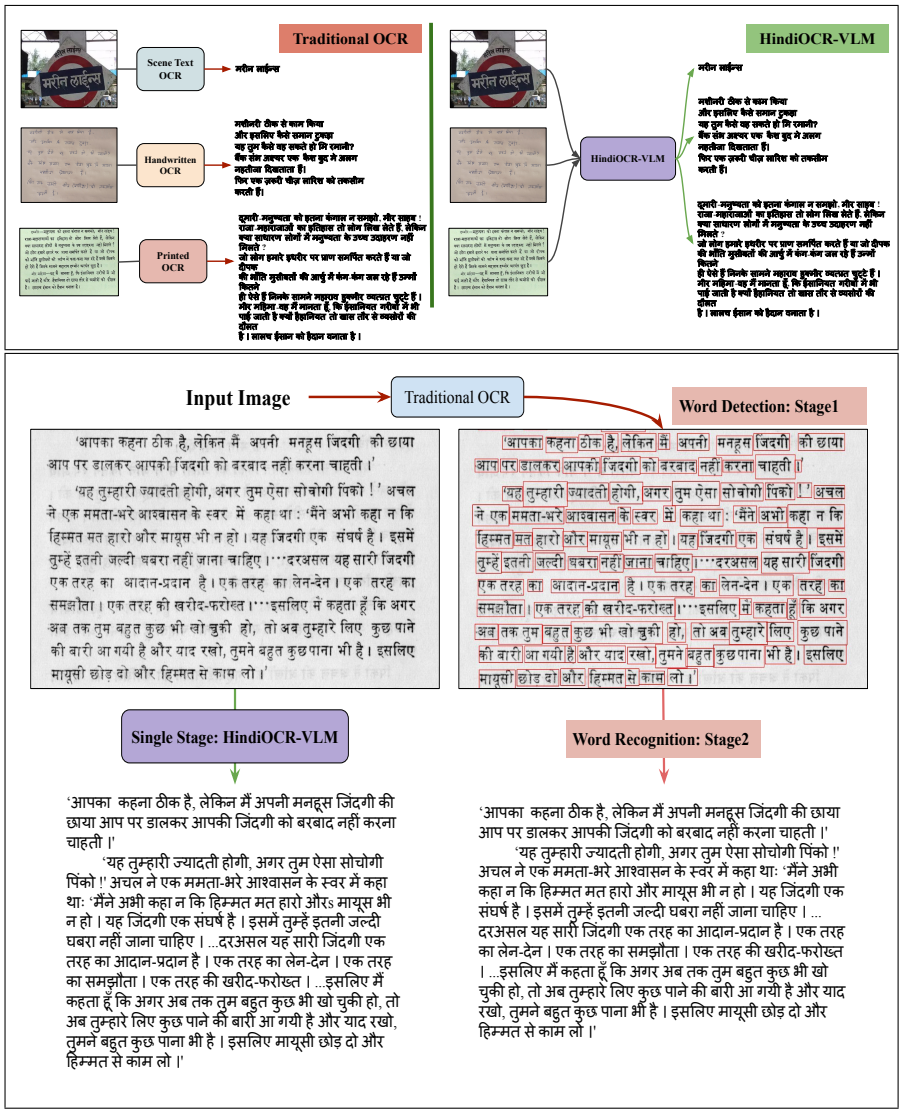

Fig. 1: Illustrates key differences between traditional OCR and *HindiOCR-VLM*: traditional methods require separate models for each modality and follow a two-stage process (localization then recognition), while VLMs handle all modalities in a single model and recognize text directly without explicit localization.

handwritten and scene text recognition. Their multimodal pre-training ensures strong generalization across fonts, writing styles, and noisy environments. Notably, VLMs excel in recognizing multilingual and code-mixed text [58], a persistent challenge in Indian OCR. Their cross-modal attention dynamically adjusts

predictions based on character shapes and linguistic context, making them highly effective for complex Indian scripts.

Indian OCR [9, 32, 39, 44] presents challenges due to diverse scripts, complex characters, and multilingual text composition. Scripts like Devanagari, Tamil, and Bengali feature ligatures and diacritics, complicating segmentation and recognition [40, 44]. Similar-looking characters, low-resolution text, and code-mixing further complicate traditional OCR approaches. Additionally, the scarcity of annotated datasets limits model generalization. Given these challenges, VLMs offer a robust solution by integrating visual and linguistic cues, excelling in recognizing occluded, degraded, and contextually ambiguous text across multiple writing styles.

We adapt Vision-Language Models (VLMs), particularly GOT OCR 2.0 [52], to develop *HindiOCR-VLM* for Hindi OCR by fine-tuning a model initially trained on Chinese and English using Low-Rank Adaptation (LoRA). Our adaptation includes two strategies: (i) a *single-stage* model that directly predicts page-level text without intermediate word or line detection, applied to printed pages, and (ii) a *two-stage* model for multi-domain settings (printed, handwritten, and scene text), which detects words and recognizes each individually. We introduce a progressive learning strategy for the *single-stage* model to enhance language learning and convergence. Leveraging the vision encoder's strong representations, our model supports effective multi-domain training. Ablation studies and evaluations show that *HindiOCR-VLM* handles Devanagari script complexities well and outperforms domain-specific and two-stage models, offering a unified and robust OCR solution.

Our contributions are summarized as follows:

- To the best of our knowledge, this is the first study to leverage modern Vision-Language Models (VLMs) for unified, *single-stage* OCR in Hindi. Fig. 1 shows the difference between traditional OCR and *HindiOCR-VLM*.
- We demonstrate that progressive learning during fine-tuning significantly improves the language learning abilities of *HindiOCR-VLM* and accelerates convergence.
- Extensive experiments show that *HindiOCR-VLM* outperforms in printed and handwritten and is comparable in scene text to industry-grade and open-source OCR models.

2 Related Works

2.1 Traditional OCR

Document imaging has progressed with advanced text detectors like EAST [62], PixelLink [15], CRAFT [5], and TextSnake [38], allowing polygonal and curved representations for better localization. Early OCR relied on handcrafted features and statistical models [8, 24], while deep learning tools such as Tesseract [50], EasyOCR [23], and MMOCR [30] use CNNs, RNNs, and attention for robust

script recognition. Handwriting remains difficult due to style variability, addressed through datasets like IAM [20] and models [12,16,36] using recurrent or transformer-based architectures [35,53].

Beyond recognition, layout analysis detects structural elements like paragraphs and tables. Tools like LayoutParser [48] and LayoutLM series [22,55,56] integrate textual, visual, and spatial cues for better document understanding, with LayoutLMv3 optimized for layout-aware tasks. Modern end-to-end models unify detection, recognition, and layout analysis to retain document semantics. With LLMs and VLMs [2,52], explicit layout detection and word cropping are increasingly unnecessary, as these models process full pages holistically — an approach we adopt to enhance page-level learning.

2.2 LVLM-driven OCR

Recent advances in Vision-Language Models (VLMs) and Large Language Models (LLMs) have improved OCR by integrating text recognition with contextual reasoning, enabling document parsing and structure extraction. Models like Donut [29], Pix2Struct [33], and OFA-OCR [37] directly generate structured outputs without traditional OCR steps. oCLIP [57] and MGP-STR [51] enhance recognition using weak supervision and linguistic integration. Multimodal transformers such as MVLT [54], CLIP4STR [60], and CLIP-LLaMA [61] further boost accuracy. Scaling VLMs [11] shows OCR improvements with larger models, while Phi-3 Vision [2] balances strong performance and efficiency with just 4.2B parameters, making it ideal for edge devices. Our ablation compares fine-tuned GOT OCR 2.0 [52] and Phi-3 Vision [2] to evaluate their capabilities in complex Hindi OCR tasks.

2.3 OCR in Indian languages

Recent advancements in Vision-Language Models (VLMs) and Large Language Models (LLMs) have significantly improved OCR for Indian languages, addressing script complexity, multilingual text, and low-resource challenges.

Pre-training in Indic languages enhances OCR by improving script handling and text normalization. IndicNLPSuite [25] provides tokenization and transliteration tools, while IndicBART [14] aids text correction and document understanding. IndicLLMSuite [26] and pre-training data and Tokenizer for Indic LLM [31] optimize tokenization and contextual understanding for complex scripts. IndicGenBench [49] benchmarks LLMs' text generation, indirectly benefiting OCR.

Domain-specific datasets further refine OCR. Paramanu [43] enhances efficiency for Indic languages, and Chitrarth [27] integrates multilingual LLMs with vision modules. Mathew *et al.* [40] use a CRNN-based model for word-level OCR in printed text. For handwritten text, Lalitha *et al.* [32] explore transformer architectures, while Lunia *et al.* [39] introduce a PARSeq-based approach for word-level scene text recognition. However, all these models remain domain-specific

and rely on external layout predictors. To address this, we aim to develop a unified model that processes entire pages holistically and generalizes across multiple domains.

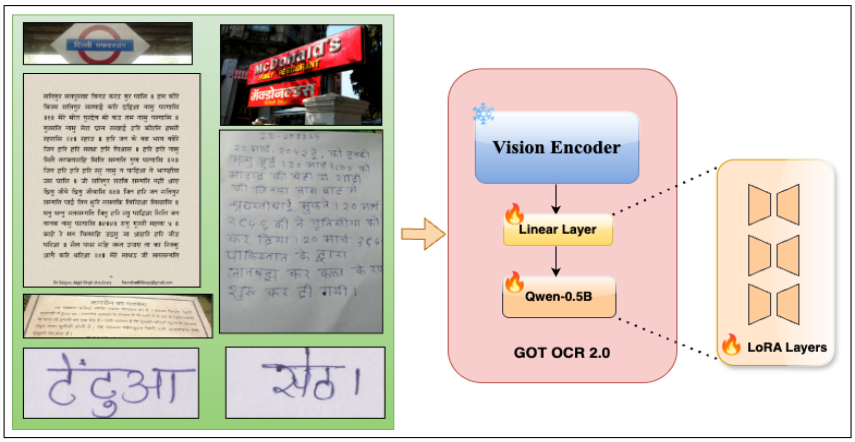

Fig. 2: Shows *HindiOCR-VLM*, an adaptation of the GOT OCR 2.0 [52] using Low-Rank Adaptation (LoRA) for the Hindi language. The vision encoder pre-trained on English and Chinese text is frozen while the decoder gets finetuned

3 Methodology and Experiments

3.1 HindiOCR-VLM

We adopt the pre-trained GOT OCR 2.0 [52] trained on a large Chinese and English corpus [6, 59]. The GOT architecture comprises three key modules: *an image encoder*, *a linear layer*, and *an output decoder*, as depicted in Fig. 2. The model's training progresses through three critical stages, beginning with the initial pre-training of the vision encoder using text recognition tasks. A compact decoder OPT-125M [59] propagates gradients while processing scene text and document-level character images. The second stage involves connecting the pre-trained vision encoder to Qwen-0.5B [6], expanding the model's capabilities through training on diverse OCR data, including intricate materials like sheet music, mathematical formulas, and geometric shapes. The final stage focuses on enhancing generalization through synthetic multi-crop and multi-page data, strategically fine-tuning the decoder while keeping the encoder frozen. We adapt this model to Hindi using LORA, finetuning the language model and keeping the vision encoder frozen. Notably, the encoder demonstrates remarkable linguistic transferability from English and Chinese to Devanagari scripts, suggesting an intriguing cross-linguistic visual compatibility in character recognition.

3.2 Model Adaptation

As shown in Fig. 2, the adaptation of the GOT model is achieved through Low-Rank Adaptation (LoRA) [21], a parameter-efficient fine-tuning method. LoRA decomposes the weight update into a low-rank approximation by introducing two matrices for each adapted weight matrix. For a given weight matrix $W \in \mathbb{R}^{d\times k}$, the LoRA update can be expressed as:

$$W + \Delta W = W + BA, \tag{1}$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ are the low-rank decomposition matrices and r is the rank of the decomposition. The effective weight update ΔW is scaled during training by a factor α/r, where α is a hyper-parameter that controls the magnitude of the update:

$$h = Wx + \frac{\alpha}{r}(BA)x. \tag{2}$$

In this implementation, adaptation is specifically applied to the decoder language model of the GOT model, while the encoder remains in a frozen state. The LoRA configuration uses an α value of 32 and a rank of 8, without any bias parameters included in the adaptation. Fine-tuning targets all linear layers within the decoder architecture, except for the language modeling head. The frozen encoder maintains the original GOT model's robust visual feature extraction capabilities, while the adapted decoder can be optimized for specific downstream tasks.

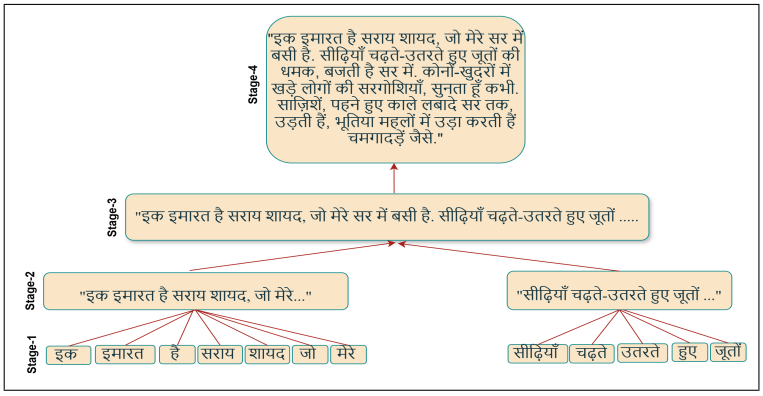

Fig. 3: Shows steps of progressive learning for language understanding in Vision-Language Models (VLMs). LoRA adaptation begins with word-level training, gradually advancing to lines, paragraphs, and full pages. This staged approach enhances accuracy on full-page text with minimal training.

3.3 Progressive Learning in Single-Stage OCR

We fine-tune the GOT OCR 2.0 model on a large corpus of printed text using a novel progressive learning strategy inspired by human language acquisition patterns. Fig. 3 demonstrates the proposed method. Rather than simultaneously presenting the model with the complete complexity of document understanding tasks, the training follows a graduated approach with distinct phases of increasing complexity. The initial phase consists of 50,000 training steps focused exclusively on word-level images. This foundational training establishes basic character and word recognition capabilities, analogous to how children learn to recognize individual words. Upon transitioning to line-level images in the subsequent phase, the model demonstrates remarkable efficiency in knowledge transfer, achieving 90% token accuracy within just 5,000 steps. This rapid adaptation suggests that word-level training creates a robust foundation for more complex text recognition tasks. Following the line-level adaptation, the training progressively introduces more complex structures, beginning with text blocks and advancing to complete pages. This incremental approach facilitates the model's capacity for direct, *single-stage* page processing during inference, thereby eliminating the necessity for explicit word or line detection. Experimental results demonstrate that progressive learning offers significant advantages over mixed training, as we later show in our ablation study.

3.4 Multidomain Learning in Two-Stage OCR

While the GOT OCR 2.0 architecture [52] demonstrates robust performance across multiple domains in its pre-trained state, its application to Indian languages presents unique challenges due to the traditional approach of using separate models for different text domains. This study proposes a unified approach to handle multiple text domains simultaneously, eliminating the need for domain-specific models in Indian language OCR systems. The training dataset is meticulously curated to ensure an equal distribution across three key domains: printed text, handwritten text, and scene text. This balanced representation ensures that the model develops domain-agnostic features while maintaining high performance across all text varieties. The printed text domain encompasses various font styles and sizes commonly found in documents and books, the handwritten domain includes diverse writing styles and variations in character formation, and the scene text domain contains text extracted from natural images with varying backgrounds, orientations, and lighting conditions. Since our dataset contains limited page-level or block-level images, especially for handwritten and scene text, we employ a *two-stage* inference approach. We first detect words using CRAFT [5] for printed and scene text, and YOLO [45] for handwritten documents. Subsequently, we feed these cropped word images into the model for recognition.

Fig. 4: Shows training samples across modalities: printed text (page, line, word) in black, handwritten words in red, and scene text words in blue.

3.5 Dataset

Our in-house printed dataset consists of 1.1M, 100K, 32K, and 3,634 word-level, line-level, block-level, and page-level images, respectively. To address the limitations posed by the insufficient availability of line-level, block-level, and page-level images, we augment these images to a large amount. We employ a comprehensive set of augmentation functions — average blur, defocus blur, Gaussian blur, glass blur, median blur, Gaussian noise, etc, resulting in 2.5M images. The augmented images introduce variability and realism into the original dataset, enhancing the trained models' robustness and generalization capabilities. For training of printed modality, we use the same in-house dataset. For mix modality training, we use handwritten images from [19] and scene text images from IndicSTR12 [39] and Bharat Scene Text Dataset [1]. Due to the lack of page, block, and line-level annotations for handwritten and scene text, progressive training was applied only to printed text. Table 1 shows dataset statistics for progressive training only for printed. For mixed modality training, we used 400K word level images per modality (1.2M total), with both setups evaluated on a challenging test set in Table 1. Fig. 4 shows a few samples of different modalities from our training set.

3.6 Training Process

The training employs Low-Rank Adaptation (LoRA) with hyperparameters $\alpha = 32$ and rank $r = 8$, enabling efficient fine-tuning while minimizing computational

Level	Train	Val
Words	11,924,480	19,654
Line	1,016,000	1,016
Block	352,737	7,054
Page	36,340	181

Modality	Pages	Words
Printed	150	47,587
Handwritten	100	4,347
Scene Text	50	378

Table 1: Presents dataset statistics, with the left table illustrating the distribution of training data for printed text, and the right table showing the composition of the testing set.

overhead. The process is distributed across two NVIDIA A6000 GPUs with a total batch size of 16 (8 samples per GPU). We use a learning rate 0.0001, Adam optimizer with 0.1 weight decay, and cosine LR scheduling. LoRA (rank 8, alpha 32) follows the original setup to match compute constraints. Progressive training starts with words, adding lines, blocks, and pages at 50k, 100k, and 200k steps, converging by 350k steps.

3.7 Evaluation Metrics

We evaluate OCR performance using two common metrics: Word Recognition Rate (WRR) and Character Recognition Rate (CRR), defined as:

$$CRR = \frac{N_p^c}{N_g^c}, \quad WRR = \frac{N_p^w}{N_g^w}, \tag{3}$$

where N_p^c and N_g^c represent the number of correctly recognized characters and the total characters in the ground truth, respectively. Similarly, N_p^w and N_g^w denote the correctly recognized words and total words in the ground truth. Higher values indicate better OCR performance.

4 Results

4.1 Comparison of OCR Methods for Printed Text

Traditional OCR models follow a *two-stage* process — detecting lines or words and sending the respective segments to a recognition model for performing OCR. In contrast, we give the entire page as input to the model, and our unified approach processes the entire text simultaneously. For a fair comparison, we present two versions of *HindiOCR-VLM*: one following a *two-stage* approach like traditional OCR systems and another *single-stage* that processes the entire page at once. As shown in Table 2, our model outperforms open-source OCR models such as CRNN [40] and Surya OCR, as well as industry-grade solutions like Google OCR and Azure OCR. Notably, it performs superiorly, requiring significantly fewer training samples than industry-grade models. Our experiments show that

[3] https://github.com/VikParuchuri/surya

Method	WRR	CRR
Google OCR	*87.35*	96.16
Azure OCR	87.17	**96.94**
CRNN [40]	87.32	*96.22*
Surya OCR[3]	83.69	80.66
HindiOCR-VLM†	84.74	92.63
HindiOCR-VLM‡	**90.77**	95.05

Table 2: Presents performance comparison with commercial and non-commercial OCRs on printed text. † and ‡ indicate *single-stage* and *two-stage* approaches, respectively. Bold and italic represent the best and second-best values, respectively.

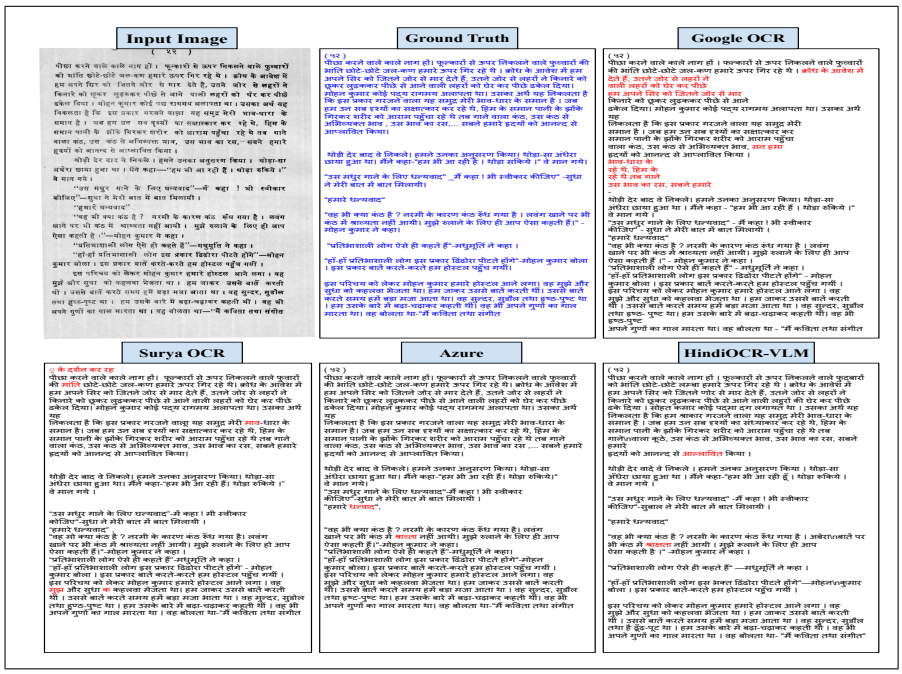

Fig. 5: Shows comparison of different OCR models on printed pages. The *single-stage HindiOCR-VLM* preserves the layout while predicting accurate text.

the *two-stage* approach significantly outperforms other baselines, while the unified approach closely matches baseline performance and produces comparable results. This performance gap arises because the *two-stage* approach eliminates background noise and spacing inconsistencies, allowing the model to focus solely on character recognition. In contrast, the unified approach must simultaneously handle text localization and recognition, making it more susceptible to errors.

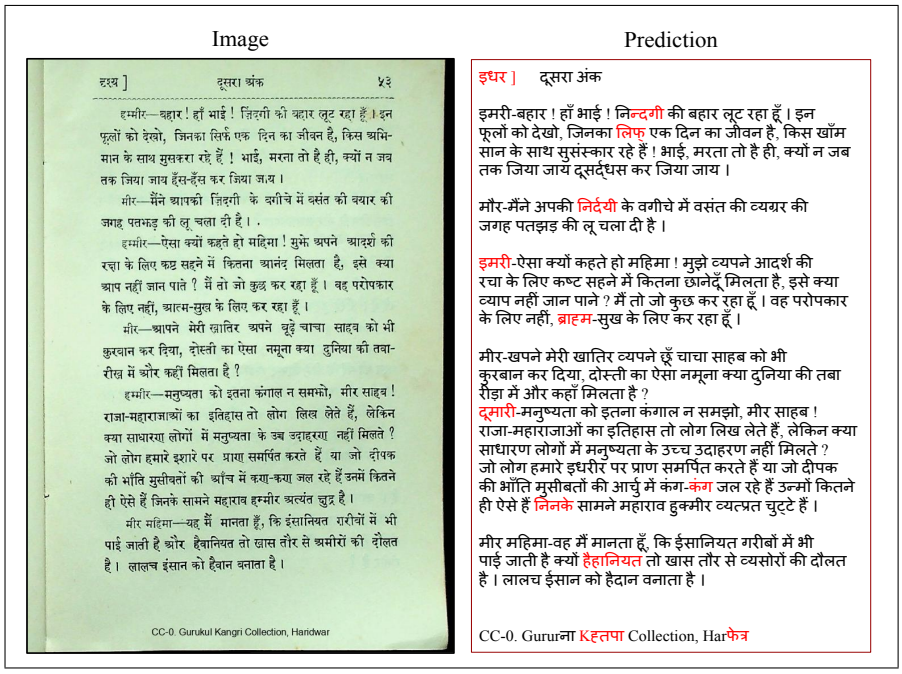

Fig. 6: Displays page-level outputs from our *single-stage* model, with the input image on the left and predicted text on the right. While trained solely on Hindi, the model successfully identifies some English text. However, its accuracy drops in lines containing both languages.

Qualitative results in Fig. 5 show that, even without intermediate detection, our model effectively preserves paragraph layouts and spacing while maintaining stable WRR compared to the industry-grade OCRs. However, the model struggles with mixed-language images, as seen in Fig. 6. We test generalization on out-of-distribution pages with variations in font colors, layouts, and print quality. We show that despite degradations of colored backgrounds, the model successfully extracts the most meaningful information.

4.2 Comparison of OCR Methods for Different Modalities

We evaluate OCR performance using both generalized models like Google OCR and domain-specific models. As discussed earlier, we adopt a *two-stage* approach in the mixed-modality setting similar to other models. We use CRNN [41], PARSeq [32], and PARSeq [39] as domain-specific models for printed, handwritten, and scene text, respectively. Here, the pipeline uses a detector at inference time. Table 4 compares different detectors, and we select the best-performing one

[4] https://lipikar.cse.iitd.ac.in/
[5] https://github.com/VikParuchuri/surya

Method	Printed Text		Handwritten Text		Scene Text	
	WRR	CRR	WRR	CRR	WRR	CRR
Google OCR	*87.35*	*96.16*	*73.27*	*80.88*	**74.38**	**87.56**
Lipikar[4]	-	-	-	-	65.81	74.16
Azure OCR	87.17	96.94	-	-	-	-
Domain Specific OCR [32, 39, 40]	87.32	96.22	69.63	73.69	*71.21*	*85.54*
Surya OCR[5]	83.69	80.66	-	-	-	-
HindiOCR-VLM	**91.28**	**97.12**	**75.21**	**89.78**	67.15	84.85

Table 3: Shows the performance comparison of OCR methods on different text modalities. We use [40], [32], and [39] for printed, handwritten, and scene text, respectively. A '-' in the table indicates that the model is not applicable for that modality.

Model	Handwritten	Scene Text	Printed
Craft [5]	0.6362	0.5576	0.5875
YOLO V11 [28]	0.9850	0.0929	0.6853
DocTR [42]	0.7444	0.1401	0.6913

Table 4: Average Precision (AP) at IoU = 0.5 for different detection models across modalities.

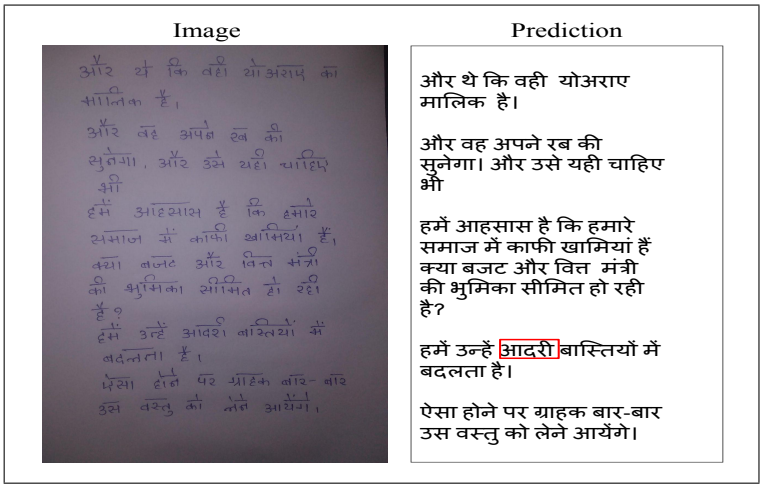

Fig. 7: Displays the performance of a mixed modality model when evaluated on handwritten images. Incorrectly predicted characters are highlighted with red color.

for each modality. With page-level annotations available across modalities, this setup enables effective multi-domain OCR through a *two-stage* pipeline that detects and recognizes words per page. Our multi-domain model outperforms all domain-specific models and surpasses industry-grade solutions across most

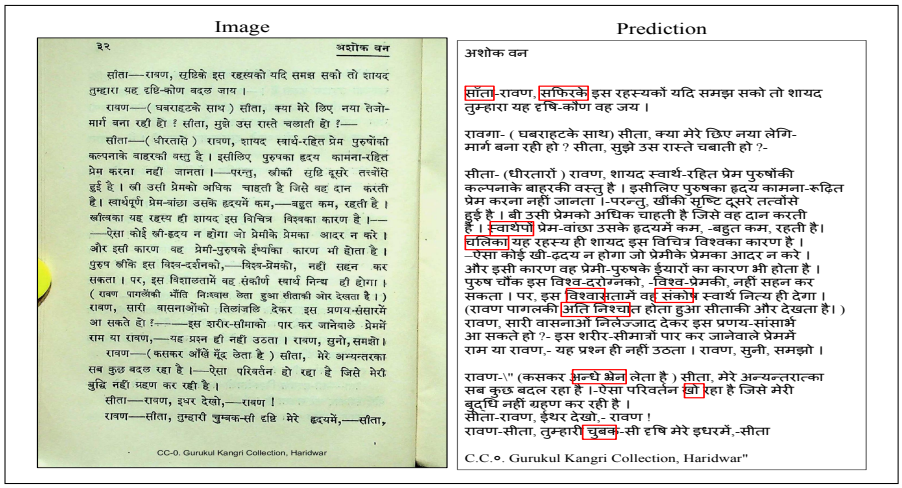

Fig. 8: Shows the performance of a mixed modality model when evaluated on printed document images. Incorrectly predicted characters are highlighted with a red rectangle.

Fig. 9: Displays the performance of a mixed modality model when evaluated on scene text images. Incorrectly predicted characters are highlighted with red color.

modalities, all while requiring significantly less training data. Quantitative results are shown in Table 3, and qualitative examples are in Figs. 7, 8, and 9. Surya OCR is trained only on printed text; hence, scene text and handwritten images are not tested on it.

Model	Training Strategy	Steps to Convergence	WRR	CRR
HindiOCR-VLM	Mixed Training	800k	82.40	90.54
HindiOCR-VLM	Progressive Learning	**350k**	**84.74**	**92.63**

Table 5: Presents a comparison of convergence speed and final accuracy on the Hindi printed test set between mixed training and progressive learning approaches. In mixed training, all image modalities are used simultaneously from the start.

As shown in Fig.7, the model handles handwritten text well despite variations in style, stroke, and spacing. Fig.8 shows similar robustness for printed text with diverse fonts and strokes. For scene text (Fig. 9), its robustness across complex backgrounds, numerals, and blur effectively. However, it struggles with irregular spacing, noisy or merged strokes, extreme angles, and punctuation errors—often due to size or resemblance to diacritics. These issues stem from the model's fine-grained feature extraction, which aids script recognition but increases sensitivity to noise, overlapping characters, blur, and perspective distortions[6].

5 Ablation Study

5.1 Impact of Progressive Learning

Progressive learning emulates the way humans learn by gradually introducing more complex training data. To assess its effectiveness, we compare it with the same base model fine-tuned from the start on a mixed dataset containing all complexity levels. As shown in Table 5, the progressive learning model converges 56.25% faster and outperforms the mixed-data model in both WRR and CRR in case of Hindi printed documents. Additionally, it shows reduced instances of hallucinations and out-of-distribution token predictions.

Distortion	Method	Sigma/S/P	WRR	CRR
Blur	Gaussian (3,3)	1	77.03	89.65
Blur	Gaussian (5,5)	1	77.07	89.80
Blur	Median (3,3)	—	79.72	91.67
Blur	Median (5,5)	—	78.97	91.16
Noise	Gaussian	5	74.01	88.10
Noise	Gaussian	10	72.54	86.12
Noise	Shot	30	75.50	86.94
Noise	Impulse	0.1	75.10	88.62

Table 6: Shows model performance on Hindi printed test set under different levels of blur and noise distortions. Here, P refers to the probability of applying the distortion, and S indicates its scale.

[6] Additional visual results are provided in the supplementary material.

Fine-tuned Model	#Step	Token Eval Accuracy
Phi-3 Vision	20k	57.03
GOT OCR 2.0	20k	**98.21**

Table 7: Present performance comparison of fine-tuned Vision-Language models for Hindi OCR recognition accuracy on Hindi printed validation set.

5.2 Model Robustness Study

We evaluate the model on Marathi, which shares the Devanagari script with Hindi but differs in vocabulary and includes additional characters. Despite these differences, the model achieves a WRR of 36.70 and CRR of 71.85 on a Marathi test set of 150 pages. While accuracy drops notably, the results demonstrate effective knowledge transfer across related languages. To further assess robustness, we add synthetic noise and blur to printed page images. As shown in Table 6, the model maintains strong performance, confirming its resilience under challenging conditions.

5.3 Effect of using Different Vision-Language Models

We evaluate the adaptability of Vision-Language Models (VLMs) for Indic OCR by fine-tuning *Phi-3 Vision* and *GOT OCR 2.0* on printed word-level images using identical training setups for 20,000 steps. As shown in Table 7, *GOT OCR 2.0* achieves a significantly higher token eval accuracy than *Phi-3 Vision*, largely due to its smaller size (580M vs. 4.2B parameters) and OCR-focused training. In contrast, the larger and more general-purpose *Phi-3 Vision*, pre-trained on diverse web data, requires more effort to adapt and converge.

6 Conclusions

This paper introduces *HindiOCR-VLM*, advancing Indian language OCR from traditional methods to Vision-Language Models (VLMs). By leveraging training strategies, we demonstrate that fine-tuning VLMs pre-trained on large-scale English and Chinese corpora significantly enhances performance in Hindi. Our results show that a unified, multi-domain approach eliminates the need for separate models for printed, handwritten, and scene text, creating a generalized OCR framework. Additionally, we highlight the potential of next-generation OCR models to perform single stage end-to-end page-level recognition seamlessly, marking a shift toward more efficient and scalable solutions.

Future work includes extending the dataset to complex multilingual documents and optimizing the model for real-time use on mobile and embedded devices through quantization and pruning. This will reduce computational load while maintaining accuracy, boosting OCR performance in diverse linguistic settings.

Acknowledgments

This work is supported by the MeitY Government of India, through the NLTM Bhashini (https://bhashini.gov.in/) project.

References

1. Bharat Scene Text Dataset. https://github.com/Bhashini-IITJ/BharatSceneTextDataset (2024) 8
2. Abdin, M., others.: Phi-3 technical report: A highly capable language model locally on your phone. ArXiv **abs/2404.14219** (2024) 1, 4
3. Adak, C., Chaudhuri, B.B., Blumenstein, M.: Offline cursive bengali word recognition using cnns with a recurrent model. In: ICFHR. pp. 429–434 (2016) 1
4. Ashwin, T., Sastry, P.: A font and size-independent ocr system for printed kannada documents using support vector machines. Sadhana **27**, 35–58 (2002) 1
5. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: CVPR. pp. 9357–9366 (2019) 3, 7, 12
6. Bai, J., others.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023) 5
7. Bazzi, others.: Ocr of degraded documents using hmm-based techniques. In: Symposium on Document Image Understanding Technology. p. 149 (1999) 1
8. Brunelli, R.: Template matching techniques in computer vision: theory and practice. John Wiley & Sons (2009) 3
9. Chaudhuri, B., Pal, U.: An ocr system to read two indian language scripts: Bangla and devnagari (hindi). In: ICFHR. pp. 1011–1015 (1997) 3
10. Chen, others .: Ocean-ocr: Towards general ocr application via a vision-language model. arXiv (2025) 1
11. Chen, others.: On scaling up a multilingual vision and language model. In: CVPR. pp. 14432–14444 (2024) 4
12. Coquenet, D., Chatelain, C., Paquet, T.: End-to-end handwritten paragraph text recognition using a vertical attention network. IEEE Trans. on PAMI **45**, 508–524 (2020) 4
13. Da, C., Wang, P., Yao, C.: Levenshtein ocr. In: ECCV. pp. 322–338 (2022) 1
14. Dabre, others.: Indicbart: A pre-trained model for indic natural language generation. arXiv preprint arXiv:2109.02903 (2021) 4
15. Deng, D., Liu, H., Li, X., Cai, D.: Pixellink: Detecting scene text via instance segmentation. In: AAAI (2018) 3
16. Diaz, D.H., Qin, S., Ingle, R.R., Fujii, Y., Bissacco, A.: Rethinking text line recognition models. ArXiv (2021) 4
17. Feng, S., Manmatha, R.: A hierarchical, hmm-based automatic evaluation of ocr accuracy for a digital library of books. In: ACM/IEEE-CS joint conference on Digital libraries. pp. 109–118 (2006) 1
18. Garain, U., Mioulet, L., Chaudhuri, B.B., Chatelain, C., Paquet, T.: Unconstrained bengali handwriting recognition with recurrent models. In: ICDAR. pp. 1056–1060 (2015) 1
19. Gongidi, S., Jawahar, C.V.: iiit-indic-hw-words: A dataset for indic handwritten text recognition. In: ICDAR. p. 444–459 (2021) 8
20. Grieggs, S., Shen, B., Li, P., Short, C., Ma, J., McKenny, M., Wauke, M., Price, B.L., Scheirer, W.J.: Measuring human perception to improve handwritten document transcription. IEEE Trans. on PAMI **44**, 6594–6601 (2019) 4

21. Hu, others.: Lora: Low-rank adaptation of large language models. ICLR **1**(2), 3 (2022) 6
22. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: LayoutLMv3: Pre-training for document ai with unified text and image masking. In: ACM MM. pp. 4083–4091 (2022) 4
23. JaidedAI: Easyocr. https://github.com/JaidedAI/EasyOCR (Year) 3
24. Jain, A., Duin, R., Mao, J.: Statistical pattern recognition: a review. IEEE Trans. on PAMI **22**(1), 4–37 (2000) 3
25. Kakwani, others.: Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In: EMNLP. pp. 4948–4961 (2020) 4
26. Khan, others.: Indicllmsuite: a blueprint for creating pre-training and fine-tuning datasets for indian languages. arXiv preprint arXiv:2403.06350 (2024) 4
27. Khan, S., Tarun, A., Ravi, A., Faraz, A., Pokala, P.K., Bhangare, A., Kolla, R., Khatri, C., Agarwal, S.: Chitrarth: Bridging vision and language for a billion people. In: ICASSP. pp. 1–5 (2025) 4
28. Khanam, R., Hussain, M.: Yolov11: An overview of the key architectural enhancements. arXiv preprint arXiv:2410.17725 (2024) 12
29. Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., Park, S.: Ocr-free document understanding transformer. In: ECCV. pp. 498–517 (2022) 4
30. Kuang, Z., Sun, H., Li, Z., Yue, X., Lin, T.H., Chen, J., Wei, H., Zhu, Y., Gao, T., Zhang, W., et al.: Mmocr: a comprehensive toolbox for text detection, recognition and understanding. In: ACM MM. pp. 3791–3794 (2021) 3
31. Kumar, others.: Pretraining data and tokenizer for indic llm. arXiv preprint arXiv:2407.12481 (2024) 4
32. Lalitha, E., Mondal, A., Jawahar, C.: Enhancing accuracy in indic handwritten text recognition. In: CVIP (2024) 3, 4, 11, 12
33. Lee, K., Joshi, M., Turc, I.R., Hu, H., Liu, F., Eisenschlos, J.M., Khandelwal, U., Shaw, P., Chang, M.W., Toutanova, K.: Pix2struct: Screenshot parsing as pretraining for visual language understanding. In: ICML. pp. 18893–18912 (2023) 4
34. Li, others.: Trocr: Transformer-based optical character recognition with pre-trained models. In: AAAI. pp. 13094–13102 (2023) 1
35. Li, others.: Htr-vt: Handwritten text recognition with vision transformer. Pattern Recognition **158**, 110967–110978 (2025) 4
36. Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: Trocr: Transformer-based optical character recognition with pre-trained models. In: AAAI. pp. 13094–13102 (2023) 4
37. Lin, J., Ren, X., Zhang, Y., Liu, G., Wang, P., Yang, A., Zhou, C.: Transferring general multimodal pretrained models to text recognition. arXiv preprint arXiv:2212.09297 (2022) 4
38. Long, S., Ruan, J., Zhang, W., He, X., Wu, W., Yao, C.: Textsnake: A flexible representation for detecting text of arbitrary shapes. In: ECCV (2018) 3
39. Lunia, H., Mondal, A., Jawahar, C.: Indicstr12: a dataset for indic scene text recognition. In: ICDARW. pp. 233–250 (2023) 3, 4, 8, 11, 12
40. Mathew, M., Mondal, A., Jawahar, C.: Towards deployable ocr models for indic languages. In: ICPR. pp. 167–182 (2025) 1, 3, 4, 9, 10, 12
41. Mathew, M., Mondal, A., Jawahar, C.: Towards deployable ocr models for indic languages. In: ICPR. pp. 167–182 (2025) 11
42. Mindee: doctr: Document text recognition. https://github.com/mindee/doctr (2021) 12

43. Niyogi, M., Bhattacharya, A.: Paramanu: A family of novel efficient generative foundation language models for indian languages. arXiv preprint arXiv:2401.18034 (2024) 4
44. Pal, U., Chaudhuri, B.: Indian script character recognition: a survey. Pattern Recognition **37**(9), 1887–1899 (2004) 3
45. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2015) 7
46. Rothacker, L., Fink, G.A., Banerjee, P., Bhattacharya, U., Chaudhuri, B.: Bag-of-features hmms for segmentation-free bangla word spotting. In: International Workshop on Multilingual OCR. pp. 1–5 (2013) 1
47. Sabir, E., Rawls, S., Natarajan, P.: Implicit language model in lstm for ocr. In: ICDAR. pp. 27–31 (2017) 1
48. Shen, Z., Zhang, R., Dell, M., Lee, B.C.G., Carlson, J., Li, W.: Layoutparser: A unified toolkit for deep learning based document image analysis. In: ICDAR. pp. 131–146 (2021) 4
49. Singh, H., Gupta, N., Bharadwaj, S., Tewari, D., Talukdar, P.: Indicgenbench: a multilingual benchmark to evaluate generation capabilities of llms on indic languages. arXiv preprint arXiv:2404.16816 (2024) 4
50. Smith, R.: An overview of the tesseract ocr engine. In: ICDAR. pp. 629–633 (2007) 3
51. Wang, P., Da, C., Yao, C.: Multi-granularity prediction for scene text recognition. In: European Conference on Computer Vision. pp. 339–355. Springer (2022) 4
52. Wei, H., others.: General ocr theory: Towards ocr-2.0 via a unified end-to-end model. ArXiv **abs/2409.01704** (2024) 1, 3, 4, 5, 7
53. Wick, C., Zöllner, J., Grüning, T.: Transformer for handwritten text recognition using bidirectional post-decoding. In: ICDAR. pp. 112–126 (2021) 4
54. Wu, J., Peng, Y., Zhang, S., Qi, W., Zhang, J.: Masked vision-language transformers for scene text recognition. arXiv preprint arXiv:2211.04785 (2022) 4
55. Xu, Y., et al.: LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In: ACL. pp. 2579–2591 (2020) 4
56. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: LayoutLM: Pre-training of text and layout for document image understanding. In: ACM SIGKDD. pp. 1192–1200 (2020) 4
57. Xue, C., Zhang, W., Hao, Y., Lu, S., Torr, P.H., Bai, S.: Language matters: A weakly supervised vision-language pre-training approach for scene text detection and spotting. In: ECCV. pp. 284–302 (2022) 4
58. Zhang, J., Huang, J., Jin, S., Lu, S.: Vision-language models for vision tasks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**, 5625–5644 (2023) 2
59. Zhang, S., others.: Opt: Open pre-trained transformer language models. ArXiv **abs/2205.01068** (2022) 5
60. 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) 4
61. Zhao, X., Xu, M., Silamu, W., Li, Y.: Clip-llama: A new approach for scene text recognition with a pre-trained vision-language model and a pre-trained language model. Sensors **24**(22), 7371 (2024) 4
62. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: East: an efficient and accurate scene text detector. In: CVPR. pp. 5551–5560 (2017) 3