Comic Generation from Hindi Short Stories

M.V. Parth
CSE Department, PES University, Bangalore, India

***Abstract*—In this paper, we present a simple yet powerful way of converting brief Hindi tales to visual comic pages. With an emphasis on Hindi only, the system is respectful of linguistic diversity and promotes native language narration. Using specially designed natural language processing approaches for Hindi, the system detects characters, divides the story into scenes, and recognizes emotions and actions. Our goal is to convert Hindi tales into complete comic strips that remain faithful to the original story but offer a tasty and lively visual spin. In the process, we make reading more enjoyable and accessible, particularly for children and all those who enjoy pictures more than simple text. This project demonstrates the ways in which AI can be leveraged creatively to make regional-language stories come alive in an innovative, contemporary manner. Keywords: Hindi stories, comic-making, visual storytelling, AI for creativity, regional language tools**

***Index Terms*—Hindi stories, comic-making, visual storytelling, AI for creativity, regional language tools**

I. Introduction

Stories have always been at the center of how people connect, learn, and share culture— particularly in India. Among the numerous languages spoken throughout the country, Hindi is one of the most popular and loved. As increasingly more storytelling goes digital, it is crucial that we develop new means of keeping such stories vibrant and relevant. Comics, with their integration of pictures and words, are a great way to accomplish this, particularly so for children and younger readers who prefer pictures to straight text.

This is a project originating from a small but significant idea: Hindi narratives need their own place in the world of artistic AI tools. Sure, there are plenty out there for English and other major global languages, but there still remains a rather large gap regarding tools that simply deal with Hindi alone. This work addresses that void by providing a system based on Hindi input only to produce comic strips—guaranteeing cultural conservation, accessibility, and creative interaction with indigenous language narratives.

II. Related Work

Existing work on comic generation and narrative has focused mostly on English as text and high-resource environments. Natural language processing, generative models, and layout algorithms have been promising, but tend to be computationally expensive and data set intensive. In addition, multilingual models tend to ignore the subtle grammatical and semantic aspects of Hindi. In contrast, our system is based on a rule-based paradigm for accuracy and interpretability. Unlike data-intensive neural networks, we use hand-designed rules to analyze Hindi text for characters, scenes, and emotions. This makes the system less heavy, simpler to deploy and better equipped to handle Hindi-language narratives, which tend to conform to tried-and-tested patterns of story and narrative flow.

III. Methodology

The system follows a modular pipeline designed to handle only Hindi input. The key stages are as follows:

A. Input Handling

The input is a short story written in Hindi. The story can be as short as a paragraph and as long as a few pages. The system assumes clear sentence structures and basic storytelling elements. Example input story - """ राम, एक सुंदर लड़का, जंगल में अपने बात करने वाले कुU◌े कालू के साथ चल रहा है। वहाँ उसकY मुलाकात सोनी नाम कY एक लड़कY से होती है। कालू कहता है 'नमस्ते'। सोनी कहती है 'हाय'। बाद में तीनों एक झील के पास बैठते हैं और बातें करते हैं। """

B. Character Extraction

Characters are identified using Hindi-specific dialogue cues such as "कहता है", "बोली", and similar expressions. Regular expressions scan the text for these patterns and extract character names. In the test story, characters like "राम", "कालू", and "सोनी" were successfully identified.

C. Scene Segmentation

The system splits the story into scenes using the Hindi full stop (।) and context-based keywords like "जंगल" (forest), "झील" (lake), etc. This allows the grouping of related actions and dialogues, forming the basis for comic panels.

D. Emotion and Action Detection

Each sentence is checked for emotional keywords (e.g., "खुश", "डर", "उदास") and verbs indicating physical position or movement (e.g., "बैठ", "चल"). This step helps determine character expressions and poses.

E. Dialogue Mapping and Comic Structure Formatting

Detected dialogues are paired with their respective speakers. Each scene is mapped into a JSON format that includes the location, involved characters, emotions, positions, and dialogues. This structured output is later passed to a visual renderer.

F. Visual Generation (External)

While this paper focuses on the text processing part, the structured data output is intended for use with text-to-image models (e.g., stable diffusion), layout engines, and bubble placement algorithms.

G. Implementation Details

The GenAI comic generation software was created in Python, taking advantage of multiple standard and third-party libraries to enable different functionalities:

Module: Used for file system operations, such as path manipulations and directory management, for compatibility with various operating systems.

json Module: Used to parse and manipulate JSON files, which contain structured data like scene information and dialogues in the hindi_story.json file.

Pillow Library (PIL):

Image: Used to open, work with, and save image files, especially for working with comic panel images.

ImageDraw: Used for drawing operations on images, like the addition of text and shapes, which is crucial for overlaying dialogues on comic panels.

ImageFont: Used for loading and using TrueType fonts, which allows for the rendering of Hindi text in the Devanagari script.

LoRA Fine-Tuning: We incorporated Low-Rank Adaptation (LoRA) for fine-tuning our base model efficiently. LoRA enables adaptation of large pre-trained models by introducing low-rank weight matrices that are trained on a task-specific dataset, reducing memory usage and improving training speed. This was especially helpful in fine-tuning our model for handling Hindi language prompts and comic-style image generation.

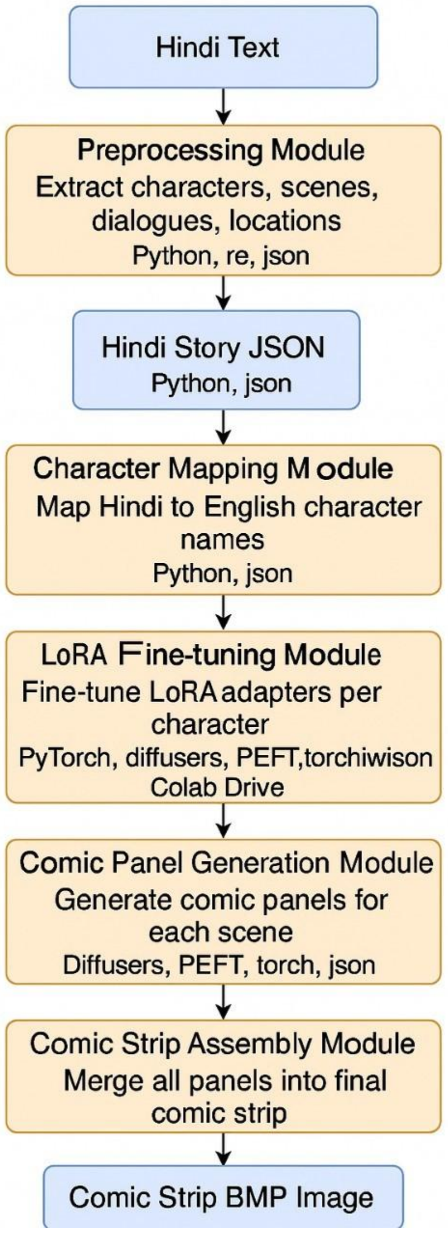
Font Integration: Noto and Noto Sans Devanagari: For displaying multilingual content, particularly Hindi, we used the **Noto** font family, which ensures visual consistency and readability across languages. Specifically, **Noto Sans Devanagari** was used to render Devanagari script accurately within both the generated images and Overleaf documentation. These fonts support full Unicode coverage and align well with our multilingual storytelling goals.

Stable Diffusion Pipeline: Our image generation relies on the **Stable Diffusion** pipeline, a state-of-the-art latent diffusion model capable of producing high-quality images from text prompts. We integrated this pipeline to convert scenes from the Hindi short story into visually compelling comic panels. The pipeline was customized and fine-tuned using story-specific prompts and LoRA weights for style control.

Google Translate Integration: To support real-time translation and multi-language compatibility, we utilized the **Google Translate API**. This was primarily used to translate Hindi sentences into English (and vice versa) to ensure semantic consistency in both the dialogue and prompts given to the image generation model. It also assisted in maintaining accurate scene descriptions when shifting between languages.

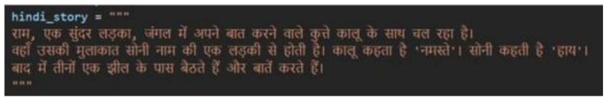
The system processes each scene by reading the image along with the related dialogues. In scenes that contain dialogues, it superimposes the text on the image via speech bubbles while maintaining proper positioning and legibility. Non-dialogue scenes are preserved in their original form. Lastly, the individual panels are joined into a complete comic strip, horizontal or vertical depending on the chosen configuration of the layout.

IV. High Level Architecture Diagram


Hindi Text
Preprocessing Module
Extract characters, scenes, dialogues, locations
Python, re, json
Hindi Story JSON
Python, json
Character Mapping Module
Map Hindi to English character names
Python, json
LoRA Fine-tuning Module
Fine-tune LoRA adapters per character
PyTorch, diffusers, PEFT,torchiwison
Colab Drive
Comic Panel Generation Module
Generate comic panels for each scene
Diffusers, PEFT, torch, json
Comic Strip Assembly Module
Merge all panels into final comic strip
Comic Strip BMP Image

V. Results

We evaluated the system on a sample Hindi story involving three characters. The story follows their interaction through various settings, including a forest and a lakeside. The character detection module identified all characters accurately. Scene segmentation yielded clear, distinct scenes, each with its associated emotional context and location. Dialogues were captured along with the speaker's emotion and position, allowing for a coherent comic panel structure. The resulting JSON file was readable, well-organized, and ready for visual generation. This proves that even simple rule-based systems can handle complex narratives when properly tailored to a specific language like Hindi.



```
hindi_story = """
राम, एक सुंदर लड़का, जंगल में अपने बात करने वाले कुत्ते कालू के साथ चल रहा है।
वहाँ उसकी मुलाकात सोनी नाम की एक लड़की से होती है। कालू कहता है 'नमस्ते'। सोनी कहती है 'हाय'।
बाद में तीनों एक झील के पास बैठते हैं और बातें करते हैं।
"""
```

Figure 1. Story input

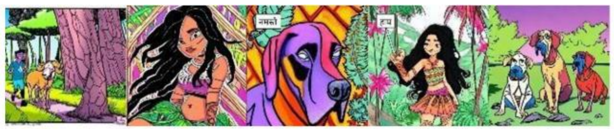


Figure 2. Comic panel generated by the model pipeline

VI. Applications

The system is versatile and can be used in a variety of contexts:

- Educational Tools: Turn textbook stories into comics to enhance reading comprehension among students.
- Children's Literature: Make Hindi stories more engaging by turning them into comics.
- Story Archiving: Preserve folktales and regional narratives visually.
- Entertainment Platforms: Allow writers and creators to turn their stories into comics without requiring graphic design expertise.

Importantly, all these applications are designed with Hindi as the only input language, ensuring focus, consistency, and cultural authenticity.

VII. Limitations

While the system performs well with short and clearly structured Hindi stories, it does come with some limitations. It currently relies heavily on straightforward sentence structures and specific keywords to detect characters, scenes, and emotions. This means that stories written in a more poetic or abstract way may not be interpreted accurately. Another key limitation is the current cap on the number of characters—right now, the system is optimized for up to three characters. If the user includes more than that, the output visuals may become inconsistent, with characters either missing, overlapping, or incorrectly placed. Additionally, since this is a rule-based system, it doesn't yet adapt to varied writing styles without manual updates. These are areas we aim to improve in future versions to make the system more flexible and robust.

VIII. Conclusion

This work presents a focused and practical system for converting Hindi stories into structured components ready for comic generation. By narrowing the scope to Hindi alone, the system ensures linguistic precision, cultural alignment, and reduced complexity. The rule-based setup we've used is intentionally simple, efficient, and easy to understand— making it a great fit for classrooms, creative spaces, and areas where access to high-end tech might be limited. Looking ahead, we're really excited about where this project can go. Right now, it's fully focused on really short Hindi stories, but there's a lot of potential to grow. We're thinking about polishing the model more so that it can take in bigger stories and give the output with more than one character talking in one frame, maybe even adding more ways for users to personalize their comics—like picking more detailed backgrounds, deciding how many panels they want, or adding more characters into a scene. We also would be looking into achieving a much better consistency of characters across each frame in the final panel. The goal is to keep things simple to use while making the storytelling experience even more expressive and fun.

References

[1] Wang, Jiahao, et al. "SpotActor: Training-Free Layout-Controlled Consistent Image Generation." arXiv preprint arXiv:2409.04801 (2024). [2] Parihar, Anil Singh, et al. "HTGAN: An architecture for Hindi Text based Image Synthesis." 2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP). IEEE, 2021. [3] Heidrich, David, and Andreas Schreiber."Visualizing source code as comics using generative AI." 2023 IEEE Working Conference on Software Visualization (VISSOFT). IEEE, 2023. [4] Alhabeeb, Sarah K., and Amal A. Al-Shargabi. "Text-to-Image Synthesis With Generative Models: Methods, Datasets, Performance Metrics, Challenges, and Future Direction." IEEE Access (2024). [5] Ramesh, Aditya, et al. "Zero-shot text-to-image generation." International conference on machine learning. PMLR, 2021. [6] Chen, Wenjuan, et al. "Developing an Intermediate Framework for Enhancing Comic Creation Through Generative AI." International Conference on Human-Computer Interaction. Springer, 2024. [7] Dylag, Jakub J., et al. "Automatic Geo-alignment of Artwork in Children's Story Books." arXiv preprint arXiv:2304.01204 (2023). [8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", 2009 IEEE

conference on computer vision and pattern recognition, pp. 248-255, 2009. [9] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, ”Bleu: a method for automatic evaluation of machine translation”, Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311-318, 2002. [10] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, ”Improved techniques for training gans”, Advances in neural information processing systems, vol. 29, pp. 2234-2242, 2016. [11] S. Parida, O. Bojar and S. R. Dash, ”Hindi visual genome: A dataset for multi-modal english to hindi machine translation”, Computación y Sistemas, vol. 23, no. 4, 2019. [12] D. Bahdanau, K. Cho and Y. Bengio, ”Neural machine translation by jointly learning to align and translate”, 2014. [13] J. Schmidhuber, ”Long short-term memory”, Neural Comput, vol. 9, no. 8, pp. 1735-1780, 1997. [14] Vaswani, Ashish, et al. ”Attention is all you need.” Advances in neural information processing systems 30 (2017). [15] Z. Yi, H. Zhang, P. Tan and M. Gong, ”Dualgan: Unsupervised dual learning for image-to-image translation”, Proceedings of the IEEE international conference on computer vision, pp. 2849-2857, 2017. [16] T. Qiao, J. Zhang, D. Xu and D. Tao, ”Mirrorgan: Learning text- to-image generation by redescription”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1505-1514, 2019. [17] E. Mansimov, E. Parisotto, J. L. Ba and R. Salakhutdinov, ”Generating images from captions with attention”, Proceedings of the International Conference on Learning Representations, 2016. [18] A. Schreiber and R. Struminski, ”Visualizing the provenance of personal data using comics”, Computers, vol. 7, no. 1, 2018. [19] B. Bach, N. Kerracher, K. W. Hall, S. Carpendale, J. Kennedy and N. Henry Riche, ”Telling stories about dynamic networks with graph comics”, Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems ser. CHI’ 16, pp. 3670-3682, 2016. [20] Benjamin Bach, Nathalie Henry Riche, Sheelagh Carpendale, and Hanspeter Pfister. 2017. The emerging genre of data comics. IEEE computer graphics and applications 37, 3 (2017), 6–13. https://doi.org/10.1109/MCG.2017.33 [21] Tiago Alves , Adrian Mcmichael , Ana Simões , Marco Vala , Ana Paiva , Ruth Aylett. Comics2d: Describing and creating comics from story-based applications with autonomous characters [22] Ying Cao , Antoni B Chan , Rynson Wh Lau. Automatic stylistic manga layout [23] Sung-Bae Cho , Kyung-Joong Kim , Keum-Sung Hwang. Generating cartoon-style summary of daily life with multimedia mobile devices [24] Gunasekara, Pramoda P., et al. ”Generate comic strips using ai.” Proceedings of Conference on Transdisciplinary Research in Engineering. Vol. 1. No. 1. 2024. [25] Terra, William Albuquerque. ”Artificial comedy: assessing the potential of generative AI in visual humour.” (2024).