ComicVerse: A Multimodal Deep Learning System for Comic Narration and Audiobook Generation

Ashutosh Mishra, Tony P Joy, Venkata Ajay Kolla, Parth Patel Pravinkumar, Prakash S and Akhzar Farhan

DA225o: Deep Learning, Summer Semester 2024-25
Indian Institute of Science, Bangalore
https://github.com/parthiisc/DA2550$_{G}rp13.git$

Abstract. Comics are a hybrid storytelling medium combining imagery and text. Despite their popularity, comics are often inaccessible to visually impaired individuals and non-native readers. We introduce ComicVerse, an AI-driven system that transforms comic PDFs into narrated stories and audiobooks using a combination of deep learning techniques. The system leverages vision-language models (GPT-4o-mini), large language models for narrative synthesis, and neural text-to-speech (TTS-1) systems to produce high-quality audio. A key novelty lies in fusing visual data (images) with OCR text to form a multimodal prompt for story generation. The system supports style control and multilingual output, and is deployed via a user-friendly Streamlit interface. Our method illustrates the integration of modern deep learning APIs into a practical, creative AI application, enabling inclusive and dynamic storytelling from static visual media.

1 Introduction

Comics blend visual artistry and written dialogue to craft immersive narratives, but this format can exclude readers with visual impairments or limited language proficiency. Traditional digital comic readers often remain static, failing to harness AI's capabilities for enhanced comprehension or interactivity. Recent advances in multimodal deep learning and neural speech synthesis, however, enable us to bridge this gap by converting visual storytelling into natural language and spoken audio.

In this work, we introduce **ComicVerse**, an end-to-end system that leverages four state-of-the-art AI components—OCR via Tesseract to extract text from images, GPT-4o-mini for multimodal page-level narration, to aggregate page summaries into a coherent short story, and TTS-1 for high-quality audiobook generation. By integrating these subsystems, ComicVerse transforms static comic pages into accessible, engaging prose and audio, broadening the medium's reach to new audiences.

2 Related Work

Multimodal models have increasingly been applied to document understanding tasks. Boukhers and Bouabdallah [2] and Audebert et al. [1] demonstrated early approaches combining visual and textual cues for metadata extraction and document classification. WuKong [5] extends this to long-form PDF reading using sparse sampling, improving efficiency in processing lengthy documents.

In the domain of comics, Sachdeva and Zisserman [4] proposed a system to generate prose from comic panels, focusing on panel-wise segmentation and text extraction. DancingBoard [3] explored motion comic generation to enhance storytelling experience.

Our work differs by focusing on full-page multimodal batching, efficient image scaling, and end-to-end story generation with translation and audio support, enabling an accessible pipeline for converting static comics into coherent narrated stories.

3 System Architecture

3.1 Pipeline Overview

ComicVerse follows a sequential pipeline:

1. **PDF to Images:** Comic PDFs are converted to compressed JPEGs for efficient processing. This step enables downstream vision models to work with individual pages as images.
2. **OCR Extraction:** Tesseract extracts embedded text from each page. OCR is crucial for capturing dialogue and narration that may not be easily interpreted visually, especially for speech bubbles and captions.
3. **Multimodal Page Narration:** GPT-4o-mini receives both the image and OCR text for each page. This fusion allows the model to reason about both visual and textual cues, resulting in richer and more accurate page-level summaries. The batch processing of up to 5 pages at a time improves efficiency and context retention.
4. **Story Aggregation:** The page-level narrations are combined and rewritten by GPT-4o-mini into a single, coherent short story. This step allows for style control (neutral, dramatic, fun) and ensures narrative flow across the entire comic.
5. **Translation:** The final story can be optionally translated to Hindi (or other languages) using GPT-4o-mini, increasing accessibility for non-English speakers.
6. **TTS Generation:** The story is converted into high-quality audio using TTS-1, making comics accessible to visually impaired users and enabling audiobook-style consumption.

3.2 Design Rationale

The architecture is designed to maximize accessibility, flexibility, and narrative quality:

- **Multimodal Fusion:** By combining OCR and image data, the system leverages both explicit text and visual context, overcoming the limitations of using either modality alone.

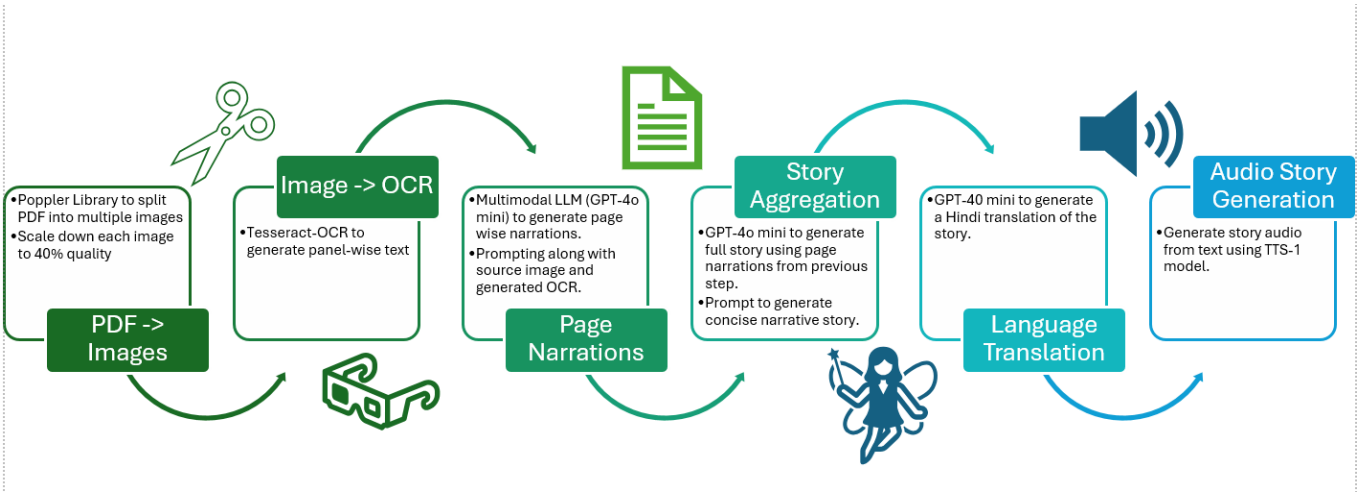

Figure 1. ComicVerse end-to-end pipeline

- **Batch Processing:** Processing multiple pages at once allows the model to maintain context and reduces API calls, which is important for long comics and cost efficiency.
- **Style Control:** Allowing users to select the tone of the generated story (neutral, dramatic, fun) demonstrates the creative potential of LLMs and increases user engagement.
- **Modular Components:** Each step (OCR, narration, aggregation, translation, TTS) is modular, making the system extensible for future improvements (e.g., panel-level narration, more languages, or custom voices).
- **User Interface:** The Streamlit app provides an intuitive front-end, enabling users to upload comics, select options, and interact with the AI-generated content, including a chatbot for follow-up questions.

3.3 Model Contributions

Component	Model	Role
OCR	Tesseract	Text extraction from images
Page Summarization	GPT-4o-mini	Image + text narration
Story Synthesis	GPT-4o-mini	Tone-controlled story generation
Speech Synthesis	TTS-1	Voice generation from text
Translation	GPT-4o-mini	English to Hindi translation

Table 1. AI Model Components in ComicVerse

3.4 Model Parameters and Prompting

Model Parameters: For page narration and story aggregation, we use OpenAI's GPT-4o-mini model with a temperature of 0.7 and a maximum token limit of 600–750 per call. The TTS-1 model is used for text-to-speech synthesis, and Tesseract is used for OCR with English as the default language.

Prompting Process: Each page is processed by sending both the compressed image (JPEG, scaled to 40 percent of original size for efficiency) and the OCR-extracted text to GPT-4o-mini. The prompt instructs the model to provide detailed, panel-wise narration, referencing both visual and textual cues. For story aggregation, the prompt includes all page narrations and a style instruction (neutral, dramatic, or fun), guiding GPT-4o-mini to rewrite the content into a coherent short story.

3.5 Image Preprocessing & Batching Strategy

To strike a balance between throughput, API cost, and downstream accuracy, we apply two complementary optimizations at the very front of the pipeline:

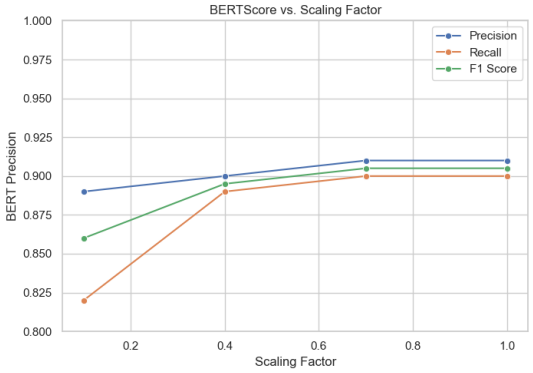

Figure 2. Processing time vs Scaling factor

3.5.1 Scaling Factor & JPEG Compression :

Each page is resized to **40 %** of its original width and height, then saved as a JPEG at `quality=60`. At this setting, file sizes drop by roughly 80 %, translating to lower upload/download latency and reduced API charges, while still preserving the fine detail needed for reliable OCR and vision-language reasoning.

3.5.2 Batch Processing :

Rather than sending pages one by one, we group up to **5 pages** into a single multimodal API call. Batching not only amortizes per-call overhead but also allows GPT-4o-mini to maintain cross-page context, improving narrative cohesion.

Scaling Factor	Images per Batch	Time per 10 Images (s)	BERT Precision	BERT Recall	BERT F_1
1.0 (100 %)	2	133	0.91	0.90	0.905
0.7 (70 %)	3	120	0.91	0.90	0.905
0.4 (40 %)	**5**	**80**	**0.90**	**0.89**	**0.895**
0.1 (10 %)	8	50	0.89	0.82	0.860

Table 2. Impact of Image Scaling on Batch Size, Processing Time, and BERTScore Metrics

Table 2 summarizes our experiments where reducing the scaling factor from full resolution down to 40 % yields a forty percent reduction in latency (133 s→80 s per 10 images) with only a minor

drop in BERT F_1 (0.905→0.895). Dropping further to 10 % gives even faster throughput but at the cost of a larger precision/recall gap. Accordingly, we adopt **40 % scaling** and **batches of 5 pages** as our default, which provides a strong middle ground:

- **Latency:** 80 s per 10 pages
- **Narrative Quality:** BERT $F_1 = 0.895$
- **API Efficiency:** 5× fewer calls and 60 % smaller payloads

4 User Features

ComicVerse is designed with accessibility and user experience in mind. The Streamlit-based web app provides the following features:

- **PDF Upload:** Users can upload any comic PDF for processing.
- **Story Style Selection:** Choose between neutral, dramatic, or fun storytelling styles for the generated story.
- **Audiobook Generation:** Instantly convert the generated story into high-quality audio.
- **Multilingual Output:** Option to translate the story into Hindi (or other supported languages).
- **Downloadable Content:** Download both the generated story text and the audiobook as files.
- **Interactive Chatbot:** Chat with the AI about the generated story, ask questions, and get contextual answers.
- **Modern UI:** Responsive, professional interface with tooltips, progress indicators, and accessibility features.

4.1 User Interface Integration

ComicVerse is delivered as a modern web application using Streamlit. The UI allows users to:

- Upload comic PDFs and select processing options (story style, language, audio).
- View the extracted story and listen to the generated audiobook directly in the browser.
- Download both the story text and audio files for offline use.
- Interact with an AI-powered chatbot to ask questions about the generated story, enabling deeper engagement and accessibility.
- Experience a responsive, accessible, and visually appealing interface with real-time progress indicators and tooltips.

The integration with Streamlit ensures that all deep learning components (OCR, GPT-4o-mini, TTS-1) are accessible through a single, user-friendly platform, making ComicVerse a true end-to-end AI application for creative media transformation.

5 Evaluation and Analysis

We evaluate three pipelines for comic-to-story generation on multiple sample comics: **(1) OCR-to-GPT**, where only text extracted via OCR is sent to GPT; **(2) OCR+Image-to-GPT**, our proposed pipeline which sends both extracted text and corresponding images to GPT; and **(3) PDF-to-GPT**, the baseline where the full comic PDF is sent directly to GPT-4.

Qualitative Analysis

To evaluate the effectiveness of our proposed pipelines, we compare them against a baseline that directly passes comic PDF pages as context to ChatGPT (Vision) for end-to-end story generation as shown in Table 3. While this baseline leverages the model's document understanding capabilities, it lacks explicit control over intermediate representations like OCR outputs or visual-textual segmentation.

The **OCR_TO_GPT** pipeline, which uses only the text extracted via OCR, produced semantically rich summaries that maintained strong alignment with ground truth (e.g., BERTScore of 0.871 and Cosine Similarity of 0.869 in Sample 1). While slightly less readable (FRE: 52.53 in Sample 1), its outputs were more grounded in the actual textual content of the comic, leading to better factual consistency.

The **Images_and_OCR_to_GPT** pipeline, which integrates both image and OCR data before passing to GPT, consistently performed well across metrics. Notably, it achieved the highest non-baseline BERTScore (0.881) and Cosine Similarity (0.885) in Sample 2, indicating better contextual comprehension. Although readability (e.g., FRE: 48.29 in Sample 2) was slightly lower, this reflects a trade-off for including richer detail and narrative coherence.

In summary, while the baseline approach benefits from fluency and high-level summarization capabilities, it risks missing nuanced content and introducing hallucinations. The proposed pipelines—especially **Images_to_OCR_to_GPT**—offer better control and grounding, resulting in more faithful and context-aware summaries of comic narratives. This highlights the value of integrating structured OCR and image information in the generation pipeline.

File	Pipeline	FRE	FKG	BERT Score	Cosine Sim.
Sample 1	OCR_TO_GPT	52.53	9.96	0.871	0.869
	Images_and_OCR_to_GPT	60.05	8.57	0.875	0.801
	Baseline (PDF_to_GPT)	62.75	8.81	1	1
Sample 2	OCR_TO_GPT	61.95	8.11	0.864	0.701
	Images_and_OCR_to_GPT	48.29	11.43	0.881	0.885
	Baseline (PDF_to_GPT)	62.75	8.38	1	1
Sample 3	OCR_TO_GPT	60.35	8.56	0.87	0.731
	Images_and_OCR_to_GPT	57.58	8.99	0.869	0.918
	Baseline (PDF_to_GPT)	67.77	7.36	1	1
Sample 4	OCR_TO_GPT	52.10	10.39	0.831	0.481
	Images_and_OCR_to_GPT	52.01	10.65	0.842	0.0.492
	Baseline (PDF_to_GPT)	67.02	7.39	1	1

Table 3. Performance of Pipelines Across Comic Files

6 Discussion and Future Work

ComicVerse demonstrates the power of integrating multimodal deep learning APIs for creative applications. Future work could include:

- Panel-level narration and character voice separation
- Support for more languages and voices
- Improved UI/UX for accessibility
- Fine-tuning models for specific comic genres
- Real-time or mobile deployment

7 Conclusion

We presented ComicVerse, a practical system that leverages state-of-the-art deep learning models to convert comics into accessible stories and audiobooks. Our approach highlights the potential of multimodal AI for inclusive storytelling and creative media transformation.

Acknowledgements

This project was completed as part of a Deep Learning course. We thank our instructor and peers for their feedback and support.

References

[1] N. Audebert, C. Herold, K. Slimani, and C. Vidal. Multimodal deep networks for text and image-based document classification. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 427–443. Springer, 2019.

[2] Z. Boukhers and A. Bouabdallah. Vision and natural language for metadata extraction from scientific pdf documents: a multimodal approach. In *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*, pages 1–5, 2022.

[3] L. Chen, S. Li, Z. Li, and Q. Li. Dancingboard: Streamlining the creation of motion comics to enhance narratives. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pages 477–503, 2025.

[4] R. Sachdeva and A. Zisserman. From panels to prose: Generating literary narratives from comics. *arXiv preprint arXiv:2503.23344*, 2025.

[5] X. Xie, H. Yan, L. Yin, Y. Liu, J. Ding, M. Liao, Y. Liu, W. Chen, and X. Bai. Wukong: A large multimodal model for efficient long pdf reading with end-to-end sparse sampling. *arXiv preprint arXiv:2410.05970*, 2024.

8 Team Member Contributions

Member	**Main Contributions**
Ashutosh Mishra	Pipeline definition, PDF to image extraction, OCR extraction, Multimodal page narration, story aggregation, chatbot integration, Translation feature
Venkata Ajay Kolla	Frontend and backend app development, Text-to-Speech model, chatbot interface, story style feature
Tony P Joy	Pipeline efficiency improvements, experiments on batching and story style generation, prompt engineering, data collection
Parth Patel Pravinkumar	Evaluation pipeline creation, metric-based analysis using cosine similarity and BERTScore
Prakash S	Evaluation design and analysis, collaborative documentation and reporting
Akzhar Farhan	UI testing and feedback, data collection, feature validation

Table 4. Role Summary of Team Members in ComicVerse Project