AI-POWERED COMIC CRAFTER SYSTEM

A Project Report
submitted in fulfilment of the
requirements for the Intel Unnati Industrial Training 2025

Bachelor of Technology
in
CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

by

KOTHAKONDA HARSHINI– 23951A6661

GOLLA JATIN – 23951A6667

Department of
CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal, Hyderabad – 500 043, Telangana

March, 2025

DECLARATION

I certify that

a. The work contained in this report is original and has been done by our team under the guidance of my supervisor(s).

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in preparing the report.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright ownersof the sources, whenever necessary.

Date: 29/03/2025

Signature of the Students

Kothakonda Harshini

Golla Jatin

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to everyone who contributed to the successful completion of this project, **"AI-Powered Comic Crafter System."**

First and foremost, I extend my heartfelt thanks to my mentor, **Dr. Khalandar Basha**, for his invaluable guidance, insightful feedback, and continuous support throughout the development of this project. His expertise and encouragement have played a crucial role in shaping my ideas and ensuring the success of this work.

I would also like to express my deep appreciation to my **Head of Department, Dr. P Ashok Babu**, for his constant motivation and support, which have been instrumental in my academic and professional growth.

A special thanks to **Intel Unnati** for providing me with this incredible internship opportunity. The experience and knowledge I have gained during this internship have been immensely enriching, helping me enhance my technical skills and problem-solving abilities. Their platform has provided an excellent environment for learning and innovation.

I am also grateful to my **family and friends** for their unwavering encouragement, patience, and belief in me, which kept me focused and determined throughout the project.

Lastly, I extend my appreciation to **everyone who supported me directly or indirectly** in completing this project. This experience has been a valuable learning journey, and I look forward to further exploring advancements in AI and education.

ABSTRACT

Comic Crafter is an innovative digital comic creation tool designed to streamline the process of making comics for artists, writers, and storytellers. It features AI-assisted illustration, automated panel layout, and customizable templates, allowing users to efficiently produce high-quality comics with minimal effort. Whether for beginners or professional creators, the platform provides a user-friendly interface that enhances creativity and storytelling.

Key features include speech bubble design, background customization, character modelling, and collaborative tools, enabling multiple users to work on a project simultaneously. The integration of cloud storage ensures accessibility across different devices, offering flexibility in the creation process. With AI-powered automation, Comic Crafter significantly reduces the time required for comic production while maintaining artistic integrity.

By bridging the gap between traditional and digital comic creation, Comic Crafter makes storytelling more accessible to a global audience. It is a valuable resource for comic enthusiasts, educators, and professionals, revolutionizing the way comics are designed and published.

Keywords: Digital comics, AI-assisted illustration, automated panel layout, storytelling, cloud storage, customizable templates, speech bubble design, character modelling, collaborative tools, artistic integrity.

CONTENTS

LIST OF FIGURES

CHAPTER 1

1.1 INTRODUCTION

The art of comic creation has evolved significantly, transitioning from traditional hand-drawn methods to sophisticated digital platforms. Comic Crafter is an advanced digital tool designed to simplify and enhance the comic-making process for both aspiring and professional creators. With a user-friendly interface and AI-powered automation, Comic Crafter allows users to bring their stories to life effortlessly, reducing the technical barriers often associated with comic production.

The platform offers a range of features, including customizable templates, automated panel layouts, speech bubble designs, and character modelling, making it an ideal choice for writers, illustrators, and storytellers. Additionally, its collaborative capabilities enable multiple users to work on a project simultaneously, fostering teamwork and creative synergy.

With the integration of cloud storage, Comic Crafter provides flexibility, allowing users to access and edit their work from any device. By combining artistic freedom with digital efficiency, Comic Crafter revolutionizes the comic-making experience, making it more accessible, efficient, and innovative than ever before.

1.2 PROBLEM STATEMENT

The comic creation industry faces challenges in producing high-quality, visually engaging content efficiently. Traditional methods require significant manual effort in storyboarding, illustration, and dialogue placement, making the process time-consuming and resource-intensive. Furthermore, existing AI-driven solutions lack real-time edge optimization and often depend on cloud-based models, leading to latency issues and accessibility limitations.

Comic Crafter AI addresses these challenges by integrating large language models (LLMs), AI-powered image generation, and automated layout techniques to streamline the comic creation process. By optimizing the system for edge devices, the project ensures fast, offline processing, making AI-driven comic generation more accessible and efficient.

Goals of the Project

- Develop an AI-powered system to generate comics with minimal human intervention.
- Use advanced LLMs to create engaging and structured comic narratives.
- Ensure real-time processing and local execution on devices like NVIDIA Jetson and Raspberry Pi.
- Utilize AI-based image generation models to maintain character and scene consistency.
- Implement automated speech bubble placement and panel structuring.
- Enable offline functionality to make AI-driven comic creation accessible without internet dependency.

1.3 OBJECTIVES

The primary objective of ComicCrafter AI is to empower users to generate high-quality, comic-style stories entirely offline using generative AI models optimized for edge devices. By removing reliance on cloud services, the platform ensures privacy, enhances responsiveness, and enables comic creation in secure or bandwidth-limited environments. This makes it ideal for both personal and professional use cases, such as education, storytelling, prototyping, and creative expression.

A key focus is delivering visually compelling and coherent comic narratives. ComicCrafter AI leverages large language models (LLMs) to transform text prompts into structured story arcs, including scene descriptions and dialogue. These are paired with diffusion-based image generation models, customized to support a range of comic art styles—from manga to Western and digital illustration—giving users creative flexibility to match their vision.

To ensure smooth performance across various devices, ComicCrafter AI is built with efficiency in mind. The platform integrates tools such as ONNX, TensorRT, OpenVINO, and GGUF to optimize inference for CPUs, GPUs, and mobile hardware. This allows the solution to run seamlessly on lightweight systems like NVIDIA Jetson, Raspberry Pi, or Mac M-series chips without sacrificing quality.

Another core objective is maintaining consistency across panels, both visually and narratively. Through techniques like LoRA fine-tuning and ControlNet, the system ensures recurring characters retain recognizable features and that panels are logically sequenced. Automated panel layout and speech bubble placement using computer vision and NLP further enhance the storytelling experience by reducing manual editing.

ComicCrafter AI is also designed with accessibility in mind, providing a lightweight and intuitive local UI via tools like Streamlit or Gradio. This interface allows users to input prompts, preview generated content, make adjustments, and export the final comic in multiple formats, including PDF and PNG. Whether for classroom storytelling, graphic novel creation, or game concepting, the platform adapts to a wide range of creative needs.

Looking ahead, ComicCrafter AI is built to support future growth. The architecture lays the foundation for additional enhancements such as mobile app integration, voice-to-comic capabilities, character style transfer, and advanced multi-panel coherence. These advancements aim to further expand the tool’s creative potential and position it as a cutting-edge solution for generative storytelling on the edge.

CHAPTER 2

2.1 System Architecture:

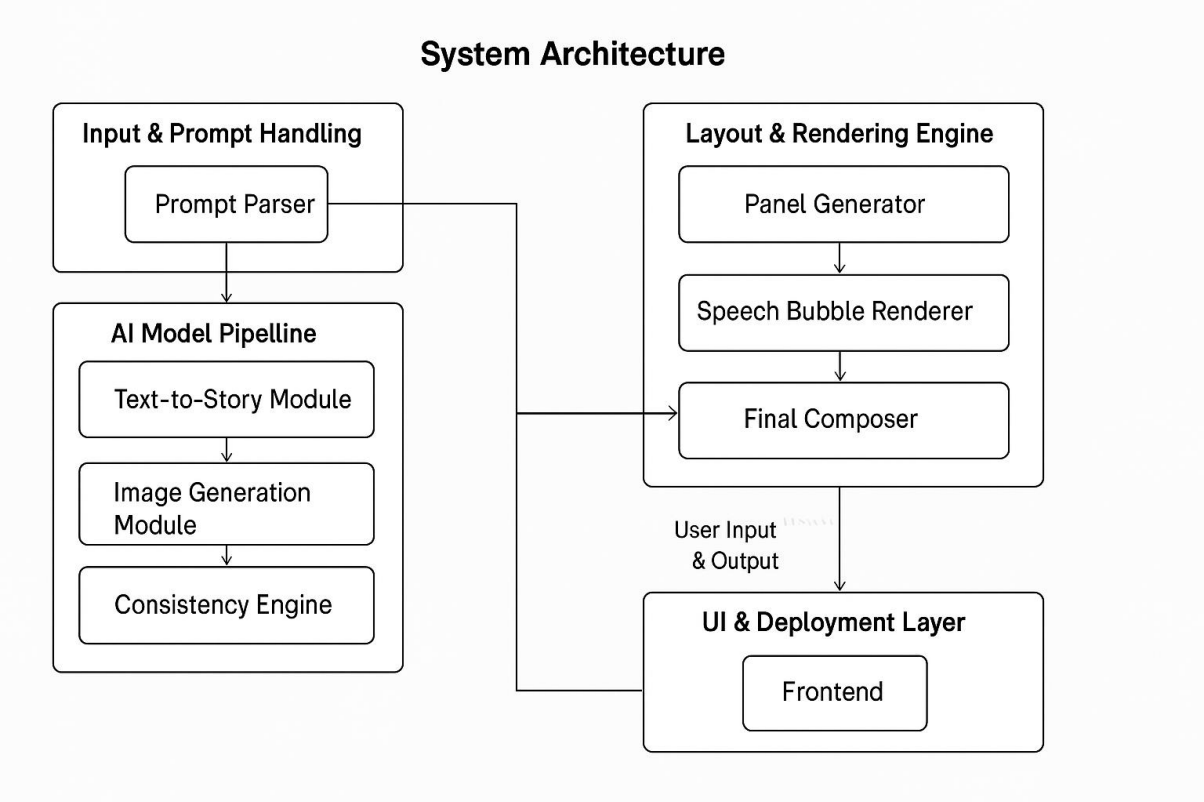

Figure 1 System Design

2.2 Requirements

➢ Hardware Requirements:

- **CPU:** Quad-core ARM Cortex-A72 (e.g., Raspberry Pi 5) or Intel i5
- **RAM:** 8 GB
- **Storage:** 32 GB SSD (for models + local cache)
- **GPU (Optional):** Integrated GPU or none (will run slower)
- **Power:** USB-C PD (if embedded system)

➢ Software Requirements:

OS:

- Linux (Ubuntu 22.04+ preferred)
- macOS (M1/M2 optimized)
- Windows 10/11 (with WSL if needed)

- Python: 3.10+
- Package Managers: pip, conda (optional), poetry (optional)

AI Models & Frameworks

- Language Model Runtime:
  - llama.cpp or ctransformers (GGUF support)
  - transformers (for fine-tuning or non-quantized inference)
- Image Generation:
  - diffusers, stable-diffusion-webui, ComfyUI (optional for advanced flows)
  - ControlNet, LoRA, xformers
- Optimization Tools:
  - ONNX Runtime
  - TensorRT (for Jetson/NVIDIA)
  - OpenVINO (for Intel)
  - GGUF for quantized LLM inference

Layout & Post-Processing

- Libraries: OpenCV, Mediapipe, matplotlib, PIL
- Speech Bubble Engine: Custom NLP + CV model or rule-based layout engine

UI & Deployment

- Frameworks: Streamlit, Gradio, or lightweight Flask-based UI
- Packaging:
  - PyInstaller (for native apps)
  - Docker (for cross-platform deployment)
  - AppImage (Linux self-contained builds)

Testing & Monitoring

- pytest, psutil, torch.utils.benchmark, and optional locust or Playwright for UI testing

CHAPTER 3

Methodology

The development of ComicCrafter AI follows a modular, system-driven methodology to ensure scalability, performance optimization, and ease of deployment on edge devices. The project is divided into distinct but interdependent stages: text-to-story generation, image synthesis, panel layout automation, and edge optimization. Each stage is guided by the principles of local execution, lightweight architecture, and user-centric design.

3.1. Text-to-Story Generation

The first step involves transforming user prompts into structured comic story elements using local large language models (LLMs) such as LLaMA 3, Mistral, or GPT-4 (quantized with GGUF). These models are fine-tuned to interpret natural language prompts and generate structured outputs containing scenes, characters, and dialogue per comic panel. The output format is typically a JSON structure that defines each panel's content, facilitating downstream integration with visual generation and layout systems.

3.2. Image Generation with Comic Styling

Once the story structure is generated, the next phase involves synthesizing visual content using Stable Diffusion XL (SDXL) models augmented by ControlNet for pose, depth, and edge control. LoRA fine-tuning techniques are applied to maintain character consistency across multiple panels. ComicCrafter AI also integrates comic-style filters using OpenCV to stylize generated images with bold lines and color grading consistent with traditional comic art. This hybrid pipeline allows the platform to adapt to different artistic styles, including manga, Western, and digital.

3.3. Panel Layout and Speech Bubble Placement

Panel construction is handled through computer vision techniques using OpenCV and Mediapipe. The system dynamically adjusts panel grids based on the number of scenes and maintains visual harmony using aspect-ratio constraints. NLP-based models determine the optimal positioning of speech bubbles based on character placement and dialogue flow, ensuring readability and visual coherence. This stage reduces manual post-processing and allows for immediate comic rendering.

3.4. Model Optimization for Edge Devices

A core methodology of the project is to deploy all AI processes locally. To achieve this, all models—both text and image—are optimized using technologies such as TensorRT (for

NVIDIA Jetson), OpenVINO (for Intel/AMD systems), and ONNX Runtime. Quantized model formats like GGUF are used for efficient LLM inference. This ensures low-latency performance on resource-constrained devices such as Raspberry Pi, Intel NUCs, and ARM-based MacBooks.

3.5. Interface and Deployment

For user interaction, lightweight local interfaces are built using Gradio or Streamlit. These interfaces allow users to input prompts, select styles, preview outputs, and export final comics in PDF, PNG, or web formats. The entire system is packaged as a standalone offline application using tools like PyInstaller or Docker for ease of installation on various platforms.

3.6. Testing and Iteration

Throughout the project, iterative testing is conducted on multiple edge devices to validate performance, latency, and output quality. Usability testing ensures that the interface is accessible to users with varying technical backgrounds. Feedback is used to refine model outputs, UI workflows, and resource usage to further align with real-world applications such as education, creative industries, and rapid prototyping.

CHAPTER 4

Model Implementation

The implementation of AI models in ComicCrafter AI is designed around modularity, local execution, and edge optimization. The system integrates quantized language models for narrative generation and diffusion-based vision models for comic-style image synthesis. Supporting models handle layout logic, character consistency, and visual post-processing. Each model is strategically selected and configured to operate entirely offline, with a focus on resource efficiency and real-time responsiveness.

4.1. Language Model (Text-to-Story Generator)

- Model Type: Quantized LLM (e.g., LLaMA 3, Mistral, GPT4All)
- Format: GGUF (for use with llama.cpp, ctransformers, or llamacpp-python)
- Function: Converts user prompts into structured comic story outputs in JSON format
- Example Output Structure:

```
{
  "title": "The Lost City",
  "panels": [
    {
      "description": "Explorer in the jungle discovers a hidden ruin",
      "dialogue": ["What is this place?"]
    },
    ...
  ]
}
```

- Offline Optimization:
    - Loaded via llama.cpp backend for low-latency inference
    - Tokenized inputs with prompt templating for few-shot learning

4.2. Image Generation Model (Text-to-Image)

- Model Type: Stable Diffusion XL (base or refiner)
- Backend: diffusers with ONNX Runtime or stable-diffusion-webui
- Support Models:

    - ControlNet (for pose, depth, edge control)
    - LoRA modules for consistent character styles
- Implementation Flow:

1. Text prompt + character description → base image
2. Apply ControlNet for pose alignment
3. LoRA fine-tunes visual style (e.g., manga, noir, watercolor)
4. Post-processing with OpenCV for comic-style line filters

- Optimization Techniques:
    - VAE precision reduction
    - ONNX export + TensorRT (for Jetson/NVIDIA)
    - Use of smaller diffusion variants for low-power devices

4.3. Character Consistency & Style Transfer

- Technique: LoRA + Token Embedding Anchoring
- Function:
    - Ensures recurring characters retain appearance across panels
    - Allows for character personalization and theme customization
- Implementation Detail:
    - Use DreamBooth or Textual Inversion during model fine-tuning
    - Reference prompts passed through memory to maintain identity
    - Dynamic swapping of LoRA weights depending on scene context

4.4. Speech Bubble Placement & Panel Layout

- Model Type: Rule-based CV-NLP Hybrid
- Libraries Used: OpenCV, Mediapipe, spaCy or transformers-based NER
- Function:
    - Analyze panel image + dialogue to determine speaker regions
    - Automatically position speech bubbles without overlapping characters
    - Layout comic panels dynamically based on content volume and pacing

4.5. Integration Pipeline

- Pipeline Orchestration:
    - Python modules orchestrate LLM → JSON → SDXL → layout → export
    - Uses a job queue system or reactive backend for stepwise inference
- UI Connection:
    - Streamlit or Gradio app calls each module through local APIs or function calls
- Export Module:
    - Outputs composite pages as PDF, PNG, or HTML comic strips

Edge Deployment Considerations

- Quantized LLMs: GGUF (4-bit, 5-bit) for small RAM devices
- Image Models: FP16 or INT8 variants with ONNX conversion
- Inference Engine: Configurable to switch between CPU, GPU, or neural accelerators depending on hardware
- Fallbacks: Lightweight fallback models for lower-end devices (e.g., TinySD or Waifu Diffusion on Pi5)

CHAPTER 5

Result

GitHub Link:

https://github.com/harshini-224/Comic-Crafter

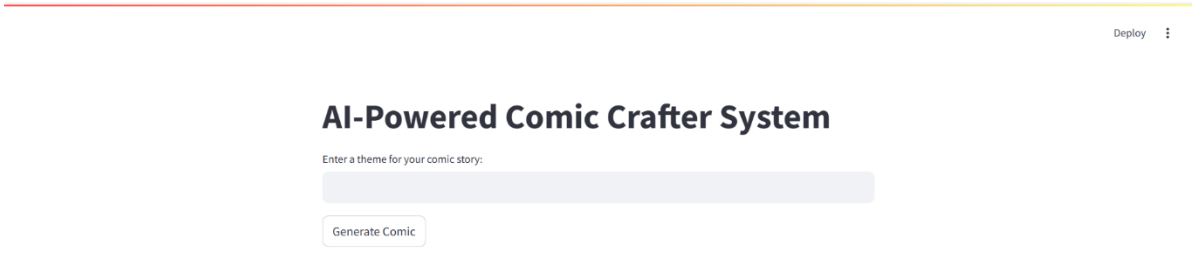

Figure 2 User Interface using Streamlit

Figure 3 Comic Strip with 4 panels

Conclusion

ComicCrafter AI successfully demonstrates the power and potential of combining generative AI with edge computing to enable fully offline, end-to-end comic creation. By leveraging quantized language models and optimized diffusion-based image generators, the system delivers a creative toolset that is both accessible and efficient. Through modular design, it supports dynamic storytelling, consistent character visualization, and automated layout—empowering users to generate comics with minimal technical expertise.

The project achieves its key goals: offline functionality, high-quality visual and narrative output, user-friendly deployment, and platform independence. The integration of speech-aware panel composition and lightweight UI further enhances its usability, especially in educational and resource-constrained settings.

Ultimately, ComicCrafter AI opens new pathways for AI-assisted storytelling, creative expression, and content generation in environments where privacy, accessibility, or connectivity are limiting factors. Its architecture provides a foundation for future enhancements, including multilingual support, user-driven character personalization, and real-time interaction—paving the way for more democratized creative tools in the AI era.