World Journal of Engineering Research and Technology

www.wjert.org

Impact Factor: 7.029 Coden USA: WJERA4

NEURO CANVAS

***[1]Jayesh Kumar Anant, [2]Hitesh Kumar Sahu, [3]Devendra Kishor Sahu, [4]Pushpa Patel, [5]Dr. Sudha Tiwari**

*[1,2,3,4]Student Scholar, Government Engineering Collage, Koni, Bilaspur.

[5]Assistant professor, Government Engineering Collage, Koni, Bilaspur.

Article Received on 05/03/2025 Article Revised on 25/03/2026 Article Published on 04/04/2026

***Corresponding Author**

Jayesh Kumar Anant

Student Scholar, Government Engineering Collage, Koni, Bilaspur.

https://doi.org/10.5281/zenodo.19435697

How to cite this Article: *[1]Jayesh Kumar Anant, [2]Hitesh Kumar Sahu, [3]Devendra Kishor Sahu, [4]Pushpa Patel, [5]Dr. Sudha Tiwari. (2026). Neuro Canvas. World Journal of Engineering Research and Technology, 12(4), 271–279.

ABSTRACT

The objective of Neuro-Canvas is to develop an offline, locally deployable AI system that integrates conversational intelligence with advanced image generation capabilities. It aims to provide users with an immersive, creative experience by enabling dynamic model swapping— utilizing an optimal Large Language Model (LLM) for text- based interactions and Stable Diffusion (via Forge UI) for high-quality artwork—while supporting distributed operation across multiple systems. The project seeks to enhance accessibility and creativity by eliminating internet dependency, offering customizable artistic outputs, and fostering iterative feedback loops for refined results.

1. SYSTEM ARCHITECTURE AND COGNITIVE INTERACTION FRAMEWORK OF NEURO- CANVAS

A. Architectural Decoupling (Modularity)

The architectural decoupling of Neuro- Canvas is achieved through a modular design that enhances flexibility, scalability, and maintainability. Key components are separated as follows:

- **LLM Module:** Handles text-based conversations using a configurable Large Language Model (e.g., llama-3.2-3b-instruct-uncensored), guided by system prompts (e.g., Artist.md), and operates independently with its own configuration (llm_config.json).
- **Forge Module:** Manages image generation via Stable Diffusion, leveraging Forge UI

settings (forge_config.json) for parameters like resolution (512x512) and artistic styles, enabling dynamic prompt processing.

- **Backend Server**: Acts as a central hub, coordinating communication between modules via dedicated ports (e.g., 6666 for LLM, 7777 for images) as defined in port_config.json, supporting distributed deployment.
- **UI/Frontend:** Provides an interactive interface for user input and output display, decoupled from backend processes, ensuring seamless integration with local or remote backend servers.
- **Configuration Files:** LLM_config.json, forge_config.json and port_config.json allow independent tuning of each module, facilitating model swapping and offline operation.

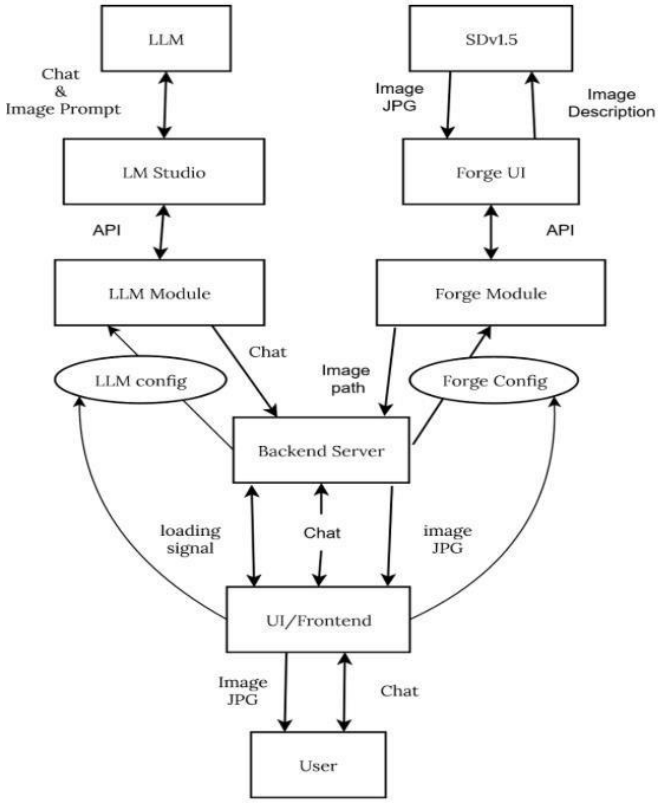

Fig. 1.1: Workflow Diagram.

This modularity enables Neuro-Canvas to run on a single system or across separate machines, with each component communicating via sockets, ensuring efficient resource utilization and adaptability for diverse creative tasks.

B. Conversational & Contextual Prompt Engineering (Cognitive)

- Neuro-Canvas uses conversational and contextual prompt engineering for enhanced user interaction.

- Features dynamic role adaptation, with the LLM adjusting tone (e.g., storytelling or roleplay) based on input.
- Intelligently extracts and enriches prompts with vivid details.
- Employs model swapping (LLM for text, Stable Diffusion for images) to optimize performance.
- Refines outputs iteratively based on user feedback for tailored, immersive results.

C. Asynchronous User Experience and Real-Time Feedback (UX)

- Neuro-Canvas delivers an asynchronous user experience through threaded operations (e.g., handle_image_generation), allowing simultaneous conversation and image generation without delays.
- Asynchronous communication via worker threads and socket listeners (e.g., ports 7777 for images, 8888 for loading) provides immediate feedback, visual cues, and rapid artwork for a seamless user experience.
- Real-time feedback is enhanced by UI updates, ensuring users see progress instantly.
- The modular design supports distributed systems, maintaining responsiveness whether running locally or across separate machines.
- User inputs trigger immediate LLM responses, with image generation running in parallel.

PROJECT DESCRIPTION

Neuro-Canvas is an offline, locally deployable AI system that integrates conversational intelligence with advanced image generation. It features a modular architecture with a Large Language Model (LLM) for text interaction and Stable Diffusion (via Forge UI) for artwork creation. The system supports dynamic model swapping—optimizing for text or image prompts—and operates across single or distributed systems (e.g., backend on one machine, frontend on another). Key functionalities include real-time feedback via threaded operations and socket listeners, iterative prompt refinement based on user input, and a seamless user experience with notifications and visual cues. It fosters creativity through immersive, tailored outputs.

A. Core Philosophical and Functional Metaphor

- **Core Philosophy:** Neuro-Canvas embodies the "Artist" metaphor, where the LLM acts as the artist's mind, interpreting user dialogue to form creative visions, while Stable Diffusion serves as the artist's hands, crafting detailed artwork.
- **Functional Metaphor:** The system mirrors a collaborative art studio—users engage in a

conversation to shape ideas, the backend orchestrates the process, and the frontend presents the evolving masterpiece, fostering an iterative creative loop.

B. Detailed Architectural Breakdown

- **LLM (Top Layer):** Drives conversational AI, using a configurable model (e.g., llama-3.2-3b-instruct- uncensored) to process text inputs and generate responses, guided by system prompts.
- **SDv1.5 (Top Layer):** Handles image generation via Stable Diffusion, leveraging Forge UI for high-quality outputs based on extracted prompts.
- **LM Studio (Middle Layer):** Manages LLM operations, providing a local interface for model execution and configuration.
- **Forge UI (Middle Layer):** Controls image generation parameters (e.g., 512x512 resolution, cinematic lighting) and integrates with Stable Diffusion.
- **LLM Module (Lower Layer):** Executes text processing and prompt extraction, communicating via port 6666.
- **Forge Module (Lower Layer):** Generates artwork based on prompts, linked to port 7777.
- **LLM Config (Support):** Stores LLM settings (e.g., base URL http://127.0.0.1:1234/v1) for dynamic adjustments.
- **Forge Config (Support):** Defines image generation settings (e.g., majicmixFantasy_v30Vae model) for tailored outputs.
- **Backend Server (Core):** Coordinates modules using socket-based communication (ports 5555 for user input, 8888 for loading), supporting distributed deployment.
- **UI/Frontend (Interface):** Offers an interactive interface for user engagement, receiving real-time updates from the backend.
- **User (Base Layer):** Initiates interactions, receiving seamless feedback.

D. Inter-Process Communication (IPC) Protocol Table 1.1 Inter process communication flow

Port(TCP/IP)	Data Flow	Purpose	Client Listener Class
Port (TCP/IP)	**Data Flow**	**Purpose**	**Client Listener Class**
5555 (user_port)	Client Server	Send user text input.	N/A (Client Server Send)
6666 (llm_port)	Server Client	Push LLM conversational reply text.	LLMListener
7777 (image_port)	Server Client	Push path to the generated image file.	Image Listener

SOFTWARE REQUIREMENT

- **Operating System:** Windows, Linux, or macOS for local deployment.
- **Python:** Version 3.9 or higher for core functionality.

- **Libraries**
  - PySide6 for GUI development.
  - OpenAI for LLM integration.
  - Pygame for audio handling.
  - Spacy for NLP processing (en_core_web_trf model).

A. Core Development and Execution Environment

The core development and execution environment for Neuro-Canvas is designed for local, offline operation:

- **Development Tools**: Python 3.9+ with PySide6 for GUI, OpenAI for LLM integration, and Requests for API handling; Spacy with en_core_web_trf for NLP, and Pygame for audio.
- **Execution Platform:** Runs on Windows, Linux, or macOS with a minimum 16 GB RAM and 8 GB GPU (CUDA-enabled) for Stable Diffusion; multi-core CPU supports threaded tasks.
- **Environment Setup:** Local host configurations for LLM and Forge UI) via port_config.json; 10 GB storage for models and outputs.
- **Runtime:** Executes asynchronously with worker threads and socket listeners (ports 5555-8888), ensuring real-time feedback and distributed capability across systems.

B. Key Software Components and Dependencies

The robust functionality relies on a well-defined set of custom classes and functions within the Python modules: Table 1.2 Components and Dependencies.

Table 1.2: Components and Dependencies.

File	Key Class/Functi on	Purpose/Fu nctional Detail
backend. py	handle_user_i nput	Main server loop; binds socket, accepts client connection, orchestrates LLM/Image.
	get_prompt	Parses LLM text response to extract the image prompt string.
	image_lock	threading.L ock ensuring serialized, safe GPU access for image generation.
llm.py	truncate_chat	Manages LLM context, maintaining token limit

	_history	(2048 default) by discarding oldest messages.
	get_soul	Dynamicall y queries the LM Studio API (/models endpoint) to fetch available LLM model names.
frontend. py	ChatUI	Primary PySide6 window, manages layout, effects, and connects all backend signals to UI slots.
	start_typing_ animation	Implements the responsive, time- delayed display of assistant replies (20ms timer).
ui_functio ns.py	LLMListener, ImageListene r, LoadingListe ner	Dedicated QThread workers for asynchrono us socket communicat ion.
	LoadingIndic ator	QWidget showing loading.gif, controlled by signals from the server.
settings_p anel.py	update_json	Universal method that instantly saves UI setting changes to disk for persistence and prepares data for server synchroniza tion.
enhancer. py	Get Detail	NLP function (currently optional) to extract scored, context-rich phrases from LLM text for prompt enrichment.

B. Client (User Interface) Hardware Justification

The Client's operational role is computationally light, ensuring broad compatibility across typical desktop and laptop configurations.

Table 1.2: Hardware.

Component	Minimum Specification
CPU/GPU	Standard Integrated or Low-Profile Dedicated
System Memory (RAM)	8 GB
Network	Ethernet/WiFi (Local Network)

HAEDWARE REQUIREMENT

- Minimum 16 GB RAM, 8 GB GPU with CUDA support for Stable Diffusion.
- Multi-core CPU for threaded operations.

A. Server (AI Processing) Hardwae Justification

The primary performance constraints in generative AI are GPU memory capacity and processing power.

ADVANTAGES

- **Offline Operation:** Functions without internet, ensuring privacy and accessibility.
- **Modular Design:** Allows distributed deployment across systems for scalability.
- **Dynamic Model Swapping:** Optimizes performance by switching between LLM and Stable Diffusion based on task.

- **Real-Time Feedback:** Provides immediate notifications and visual cues via asynchronous threads.
- **Customizability:** Supports tailored prompts and configurations for diverse creative outputs.
- **Resource Efficiency:** Leverages local hardware (e.g., GPU, multi-core CPU) for cost-effective processing.

APPLICATION

- **Creative Content Generation:** Our platform excels in generating diverse creative content, making it ideal for:
  - **Digital Artists:** Rapidly prototype visual concepts, experiment with styles, and generate artwork based on conversational prompts.
  - **Marketing & Advertising:** Create unique visual assets for campaigns, social media, and presentations with speed and efficiency.
  - **Game Development:** Generate textures, character concepts, environmental art, and UI elements.
  - **Fashion Design:** Visualize new clothing lines, patterns, and design concepts.
- **Educational Tools:** Neuro-Canvas can serve as an interactive learning tool for:
  - **Art & Design Students:** Explore the fundamentals of visual composition and experiment with different artistic styles.
  - **Creative Writing:** Visualize scenes and characters described in text, enhancing the storytelling process.

- **Personal Expression & Hobbies**
  - **Custom Artwork:** Users can create personalized artwork for personal use, gifts, or home decor.
  - **Storyboarding:** Quickly generate visual representations for narratives, film concepts, or presentations.

FUTURE SCOPE OF THE PROJECT

- **In Terms of Development**
  - **Enhanced Communication**: Replace sockets with a modern protocol for low-latency, scalable data exchange.
  - **Better Image Generation:** Add advanced techniques for faster, higher-quality and better outputs

- **Optimization:** Use memory-efficient methods and dynamic adjustments to reduce VRAM usage (6GB) and speed up iterations.
- **TTS Integration:** Upgrade to advanced offline speech synthesis for natural, customizable audio.
- **AI Personalization:** Enhance adaptive learning for user preferences.
- **Hosting Backend Online:** Provide an option to host the backend on a cloud server for enhanced accessibility and scalability.

- **In Terms of Potential Opportunities**

- Educational Use: Integrate into academic curricula for AI and art courses, enhancing student creativity.
- Commercial Applications: Offer as a tool for artists, marketers, or content creators needing custom visuals.
- Community Engagement: Build a user community for sharing prompts and artwork, boosting adoption.
- Research Potential: Serve as a platform for AI art and NLP research, attracting academic interest.

CONCLUSION

Neuro-Canvas represents a pioneering offline AI art tool, blending conversational intelligence with advanced image generation through its modular, scalable architecture. With dynamic model swapping, real-time feedback, and future enhancements like cloud hosting and mobile support, it offers vast potential for education, commerce, and research. It stands poised to redefine creative expression with its innovative approach.

REFERENCES

1. LM Studio Documentation. (2025). Local LLM inference for Windows/Linux/macOS. Retrieved from https://lmstudio.ai/docs/app
2. Forge WebUI (Stable Diffusion). (2025). GitHub repository – lllyasviel/stable-diffusion-webui-forge. Retrieved from https://github.com/lllyasviel/stable-diffusion-webui-forge
3. Python Socket Programming : Python Socket Programming Tutorial.
4. Hugging Face. (2025). Model hub for Llama-3.1, Gemma-2, Flux.1, SDXL, and LoRA collections. Retrieved from https://huggingface.co
5. Civitai. (2025). Community-driven Stable Diffusion model and LoRA sharing platform. Retrieved from https://civitai.com

6. Python GUI Tkinter : Tkinter Course - Create Graphic User Interfaces in Python Tutorial
7. Python GUI Customtkinter : https://customtkinter.tomschimansky.com/documentation/