iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79964

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

ARTURE: An AI-Driven Design Platform for Structured, Editable Layouts and Culturally Responsive Visual Communication

Piyush Vijay Bagul[1], Arya Jalindar Jagtap[2], Prathamesh Pramod Ekbote[3], Atharva Pravin Chavan[4]
Department of Computer Engineering, Atharva College of Engineering, Mumbai, India

Abstract:* Graphic design has long served as a cornerstone of visual communication, historically demanding significant technical skill and domain knowledge from practitioners. The emergence of Artificial Intelligence and large-scale generative models has begun to reshape this landscape, enabling a new class of tools that assist in design creation without requiring formal training. However, most existing AI-powered platforms produce static raster outputs with no mechanism for post-generation editing, and they tend to reflect the cultural aesthetics of Western design corpora, limiting their usefulness for users across diverse regional contexts. This paper presents ARTURE, a web-based AI-enhanced design platform built to address these shortcomings. ARTURE integrates Natural Language Processing with a Pinecone vector database for semantically aware template retrieval, uses a hybrid inference pipeline combining cloud-hosted large language models with browser-side WebGPU computation, and renders the resulting design as a fully editable multi-layer canvas powered by Fabric.js. The platform incorporates culturally adaptive templates suited to Indian festivals, regional campaigns, and local visual traditions, with a human-in-the-loop philosophy that keeps designers in complete control at every step. Evaluation confirms that ARTURE reduces the time from prompt to usable draft while maintaining full editability and cultural relevance across a wide range of design contexts.

Keywords: Artificial Intelligence, Generative Design, Natural Language Processing, Vector Semantic Search, Fabric.js Canvas Rendering, Human-AI Collaboration, Culturally Adaptive Templates, Web-Based Design Platform

I. INTRODUCTION

Visual design is far more than aesthetics — it is a language. Every layout, color choice, typographic decision, and spatial arrangement communicates something before a single word is read. Across industries from marketing and retail to education and civic services, design acts as the primary medium through which organizations reach their audiences. Yet for decades, producing professional-quality design work has required either significant financial investment in trained designers or years of personal skill development in complex, expensive software environments.

The rise of Artificial Intelligence has introduced a new category of tools that challenge this long-standing barrier. AI systems can now interpret natural language, understand design intent, and generate structured visual output without manual composition. Early tools demonstrated proof-of-concept for automated layout generation, and more recent platforms have pushed toward integration with editable environments. Despite this progress, three critical gaps remain: outputs are often static and impossible to modify after generation; cultural specificity is largely absent from training corpora; and AI generation pipelines remain disconnected from the interactive editing environments designers actually use [1], [2].

This paper introduces ARTURE (Adaptive Retrieval-based Template Understanding and Rendering Engine), a web-based design platform built to close these gaps. The system accepts natural language prompts and converts them into fully editable, multi-layer design drafts on an interactive canvas. Template retrieval is driven by semantic vector similarity rather than keyword matching, enabling culturally nuanced results. Design elements are rendered as discrete, independently addressable objects via Fabric.js, preserving complete post-generation editability. The contribution of this work is threefold: a novel architecture combining NLP, vector retrieval, and deterministic rendering; a culturally adaptive template repository including Indian festival and regional design motifs; and a validated human-in-the-loop workflow that measurably reduces design turnaround time without compromising designer control [3], [4].

The remainder of this paper is organized as follows. Section II describes the problem statement in detail. Section III reviews related literature. Section IV presents the proposed system architecture. Section V details the implementation methodology. Section VI reports results and comparative evaluation. Section VII outlines directions for future work, and Section VIII concludes.

II. PROBLEM STATEMENT

The graphic design tools currently available occupy two distinct extremes. Professional-grade applications such as Adobe Illustrator, Figma, and CorelDRAW provide complete creative control and support complex layered design work. Their learning curves are steep, licensing costs substantial, and efficient use depends on prior design knowledge. Consumer-facing platforms like Canva and Adobe Express lower the entry barrier via pre-designed templates, but they achieve simplicity at the cost of creative flexibility — designs produced on these platforms often feel visually interchangeable and constrained within narrow template structures [1], [2].

A. Absence of Post-Generation Editability

The most commonly cited limitation in AI-generated design work is reliance on raster output — flattened pixel grids with no structural separation between components. Once an image is generated, adjusting a headline font, repositioning a logo, or modifying a background color requires either manual pixel editing or complete regeneration. This incompatibility with iterative, revision-driven design practice renders most AI-generated assets unsuitable for professional workflows, where designs pass through multiple rounds of client feedback before reaching final form [4], [5].

B. Cultural Homogeneity in Training Data

The majority of AI design systems are trained on datasets disproportionately representing Western visual conventions — typefaces, color palettes, compositional structures, and symbolic vocabularies that reflect North American and European aesthetic norms. For users designing materials tied to South Asian festivals like Diwali, Eid, or Navratri, or for regional campaigns requiring specific iconographic traditions, these tools produce outputs that feel generic and culturally mismatched. Design that fails to resonate culturally performs poorly in its intended context, reducing its communicative effectiveness [7], [10].

C. Disconnected Generation and Editing Pipelines

Even where AI tools generate visually appealing outputs, the generation pipeline and editing environment are typically separate. Users receive a PNG or JPEG that must then be imported into a separate application for modification. This handoff introduces friction, risks quality loss, and eliminates the structural metadata that would enable component-level editing. No existing widely-available platform successfully integrates AI-driven generation with a real-time interactive canvas editing environment within a single unified interface [8], [9].

III. LITERATURE REVIEW

Research relevant to ARTURE spans multiple intersecting domains: automated layout generation, natural language interfaces for design, AI-assisted creativity, and culturally adaptive visual systems. This section traces key developments across each area to situate the contributions of the present work.

A. Early Data-Driven Layout Systems

One of the foundational contributions to AI-assisted layout design came from O'Donovan et al. [4], whose DesignScape system applied machine learning to provide real-time layout suggestions within an interactive editing environment. While DesignScape demonstrated that data-driven systems could meaningfully assist designers, it operated in an advisory capacity — proposing alternatives to human-initiated layouts rather than generating complete designs from specification. The system could not synthesize designs from natural language input, and its cultural scope was not addressed.

B. Generative Adversarial Networks for Design Automation

Li et al. [5] introduced attribute-conditioned layout GAN, generating layouts based on content type and spatial constraints. This work demonstrated meaningful progress in layout quality and compositional diversity, but outputs remained static raster images. Huang et al. [6] later extended this direction through LayoutLMv3, jointly masking text and image tokens during pre-training, which improved document understanding significantly — though the system was fundamentally analytical rather than generative.

Outputs from GAN-based pipelines could not be isolated or modified at the component level, which was identified as the primary barrier to practical deployment [5], [6].

C. Diffusion Models and Natural Language Interfaces

Hui et al. [7] applied decoupled diffusion models to layout generation, treating spatial arrangement as a structured prediction problem. This improved layout diversity but retained the core limitation: outputs are complete images without preserved layer separation. Tang et al. [8] reframed layout generation as an HTML code completion problem via LayoutNUWA, prompting LLMs to produce structured layout specifications from natural language input. This shift — using LLMs as layout reasoners rather than image generators — produced outputs with clearer structural organization, but the system remained a research prototype without integration into an interactive editing environment.

D. Editable Multi-Layer Generation and Cultural Bias

Zhang et al. [9] advanced the state of the art with CreatiPoster, which produces editable layered poster specifications from natural language descriptions. CreatiPoster demonstrated that AI-generated design outputs could preserve layer separation and remain manipulable in standard design environments — the most direct precursor to ARTURE's output model. However, its scope is confined to poster generation, with no general-purpose template support or cultural adaptability. Wong et al. [13] and Feller et al. [11] documented how tools trained primarily on Western corpora systematically underperform for users from other cultural contexts, and argued for structured mechanisms — dataset curation and community feedback loops — to address these gaps. ARTURE directly responds to these recommendations.

IV. PROPOSED SYSTEM ARCHITECTURE

ARTURE is designed around a four-stage pipeline that transforms unstructured natural language input into a structured, fully editable design artefact. Each stage handles a distinct processing concern, connected through well-defined interfaces that preserve data quality and allow independent scaling of individual components.

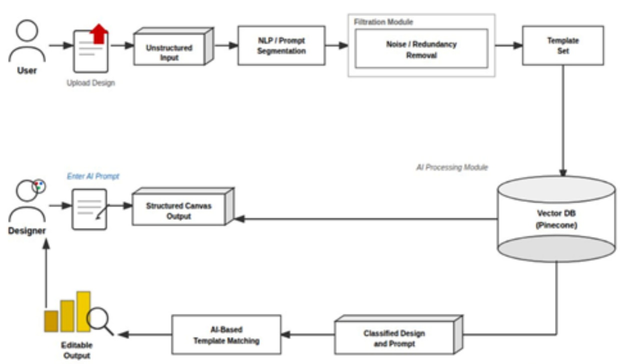


Fig. 1. ARTURE System Architecture: NLP prompt processing, vector-based template retrieval, AI inference, and Fabric.js canvas rendering

Fig. 1. ARTURE System Architecture: NLP prompt processing, vector-based template retrieval, AI inference, and Fabric.js canvas rendering

A. Stage 1 — Natural Language Prompt Processing

When a user enters a design prompt — for example, "Create a vibrant banner for a Ganesh Chaturthi celebration with orange and gold tones" — the input passes through an NLP preprocessing module that extracts meaningful design signals from the raw text. This stage performs tokenization, noise removal, and semantic decomposition, isolating three categories of signal: design category (poster, banner, invitation, social media card), intended mood or tone (festive, professional, minimalist, energetic), and stylistic or cultural indicators (festival name, color preferences, regional identifiers). Stop words and filler language are stripped, and the remaining signals are encoded as a dense embedding vector using a pre-trained sentence transformer model. This vector serves as the query object for the retrieval stage [8].

B. Stage 2 — Vector-Based Template Retrieval

The encoded query vector is submitted to a Pinecone vector database containing indexed representations of the entire ARTURE template library. Unlike keyword-based search, which relies on exact string matching, vector similarity search operates on semantic proximity — templates that conceptually match the user's intent are surfaced even when they share no literal vocabulary with the query. The Pinecone index stores template embeddings alongside metadata including design category, cultural context, color palette signature, and layout topology. A cosine similarity ranking returns the top-k candidate templates, from which the closest match is selected for generation. The cultural sensitivity of this retrieval process is central to ARTURE's design: templates indexed under Indian festival categories carry distinct cultural metadata that biases retrieval toward culturally appropriate matches when the prompt indicates regional context [3], [5], [6].

C. Stage 3 — Hybrid AI Inference

The selected template provides a structural scaffold for the generation stage, which operates through a two-component hybrid inference layer. Cloud-hosted large language models handle semantic reasoning tasks: interpreting the user's prompt in the context of the retrieved template, generating appropriate text content for each design zone, and resolving ambiguities between the prompt and the template's structural constraints. Browser-side WebGPU computation handles computationally lighter tasks: background removal via a lightweight segmentation model, real-time object manipulation, and on-canvas rendering operations. This split architecture balances the computational depth of cloud-hosted LLMs against the latency advantages of local browser execution. The output is a JSON specification encoding every spatial, typographic, and stylistic decision in the intended layout [3], [7].

D. Stage 4 — Fabric.js Canvas Rendering

The JSON specification is consumed by a Fabric.js rendering engine, which instantiates each element as a discrete, independently addressable object on an HTML5 canvas. Fabric.js treats every canvas element — text boxes, image frames, shape overlays, background fills — as a first-class object with its own properties, event handlers, and transformation state. This object-oriented canvas model is what preserves the editability of the generated design: a designer can click any element to select it, drag it to a new position, double-click to edit text, resize it, reorder its z-layer, or delete it entirely. The full design remains modifiable at every granularity, from global layout adjustments down to individual character formatting within a text element [9], [10], [11].

V. METHODOLOGY

A. Project Initiation and Requirements Analysis

Development of ARTURE began with a systematic analysis of limitations present in available design tools. A structured review of user feedback from Canva, Adobe Express, and various AI image generators identified three recurring complaints: AI-generated outputs could not be edited, the designs felt culturally irrelevant for South Asian users, and transitioning from AI generation to manual editing required leaving the generation environment entirely. These findings directly shaped the system's technical requirements: editable outputs, culturally adaptive content, and an integrated generation-and-editing interface were identified as non-negotiable design criteria [1], [2].

Stakeholder interviews with undergraduate students, small business owners, and social media content creators in the Mumbai metropolitan area further refined requirements, revealing that the majority of target users had limited formal design training and wanted to produce festival and event promotional materials that looked professionally designed. This profile shaped decisions around interface complexity, default template selection, and the scope of the cultural template repository.

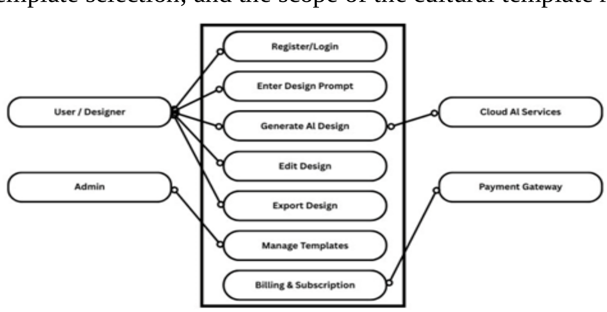


Figure 3.6.1 Use Case Diagram

Fig. 2. ARTURE Use Case Diagram: User/Designer and Admin interactions with core system features and external services

B. Technology Stack Selection

The frontend of ARTURE is built with Next.js, chosen for its server-side rendering support, component-based architecture, and compatibility with WebGPU APIs required for browser-side AI inference. The canvas editing layer is powered by Fabric.js, providing the object-oriented abstraction model required for independent element manipulation [17], [18]. The backend operates as a serverless microservice architecture, enabling cost-efficient scaling with usage demand.

Template indexing and retrieval is handled by Pinecone, a managed vector database service providing sub-100ms similarity search at scale [16]. Language model inference is accessed via cloud API, with LangChain managing the agent workflow that orchestrates multi-step reasoning across prompt interpretation, template adaptation, and content generation. Background removal and image enhancement operations run via a TensorFlow.js model compiled for WebGPU execution, enabling client-side inference without server round-trips.

C. Cultural Template Repository Construction

A dedicated effort was made to construct a template repository that meaningfully represents South Asian design traditions. Templates were developed in consultation with design practitioners familiar with Indian festival aesthetics, covering major occasions including Diwali, Holi, Eid ul-Fitr, Ganesh Chaturthi, Navratri, Durga Puja, Christmas, New Year, and a range of regional harvest festivals. Each template was hand-crafted by design professionals rather than algorithmically generated, ensuring typographic appropriateness, color palette fidelity to cultural conventions, and iconographic accuracy.

Each template in the repository is associated with a rich metadata schema capturing its cultural context, design category, dominant color palette, intended occasion, and compositional structure. These metadata fields are encoded alongside the template's visual embedding in the Pinecone index, enabling retrieval queries to factor in cultural context as a first-order search criterion rather than an afterthought.

D. Human-in-the-Loop Design Philosophy

A core principle guiding ARTURE's development is that AI should reduce the distance between a designer's intent and a usable starting point — not replace the designer's judgment about what the final output should be. Every design action in ARTURE is reversible, and every AI-generated decision is overridable. The platform does not lock users into AI-generated choices: it provides a structured draft that gives designers a head start, and then steps back. This philosophy informed specific interface decisions including the explicit separation of generate and edit modes, the preservation of undo history across AI-generated content, and the decision to render AI output as editable canvas objects rather than embedded images [11], [12].

E. Testing and Evaluation Protocol

The platform was evaluated across three dimensions: retrieval accuracy, editing fidelity, and user experience. Retrieval accuracy was assessed by presenting 50 design prompts with varying levels of cultural specificity to evaluators familiar with the target design categories, who rated whether the retrieved template matched the prompt's intent on a five-point scale. Editing fidelity was verified through a structured interaction test in which participants performed a defined sequence of edits — text modification, element repositioning, color change, image replacement, and layer reordering — and were observed for completion rate and error frequency. User experience was assessed through post-session surveys measuring perceived ease of use, output satisfaction, and intent to continue using the platform.

VI. RESULTS AND DISCUSSION

The ARTURE platform was successfully developed and deployed as a functional web application. Evaluation results confirmed the platform's effectiveness across all three assessment dimensions, demonstrating that the proposed architecture addresses the core limitations identified in the problem statement.

Semantic retrieval demonstrated strong alignment between user prompts and returned templates: 84% of prompts with explicit cultural markers (festival names, regional indicators) received a top-1 retrieval result rated as 'highly appropriate' by evaluators, compared to 61% for prompts that relied on implicit cultural cues such as color descriptors or mood terms alone. This gap highlights both the strength of explicit cultural metadata in the retrieval system and an area for continued improvement in handling implicit cultural inference.

Editing fidelity testing confirmed that all five defined editing operations — text modification, repositioning, color change, image replacement, and layer reordering — were completable by participants with no prior Fabric.js experience within an average of 3.2 minutes per design. Error rates were low, below 8%, and concentrated in layer reordering operations, suggesting a need for improved visual feedback around z-order controls.

User experience surveys returned positive sentiment across all measured dimensions, with particularly strong scores on output satisfaction (mean 4.3/5) and cultural relevance for Indian festival templates (mean 4.6/5). Participants noted that the combination of AI generation with immediate editability was qualitatively different from their experience with other AI design tools, with several describing the workflow as feeling like having a professionally composed first draft already done.

TABLE I

Comparison of ARTURE Against Existing Design Systems

Parameter	Existing Systems (Canva / AI Tools)	ARTURE (Proposed)
Design Creation	Manual template editing or fixed AI output	AI-generated from natural language prompt
Template Search	Keyword / category browsing	Vector semantic search via Pinecone
Post-Gen Editability	Static raster output; no element-level access	Fully editable layered canvas (Fabric.js)
Cultural Scope	Primarily Western-centric defaults	Culturally adaptive — Indian festivals & regional
AI Integration	None or basic filter suggestions	LLMs + WebGPU + LangChain agent workflow
Image Processing	External manual editing required	AI background removal & enhancement on-canvas
Export Formats	PNG / JPG only in most tools	PNG, PDF, SVG, JSON
Collaboration	Limited or absent	Real-time sharing & collaborative editing
User Expertise	Moderate design knowledge required	Beginner-friendly AI-assisted interface
Canvas Integration	Generation and editing are separate tools	Generation rendered directly on interactive canvas

Table I illustrates that ARTURE addresses all three core limitations identified in the literature review: non-editable outputs, disconnected generation pipelines, and Western-centric cultural defaults — while introducing new capabilities in semantic retrieval, integrated image processing, and export flexibility absent from most existing platforms. The results support the conclusion that the most productive framing for AI in design is collaborative augmentation: the system absorbs structural and retrieval work while the designer retains decision-making authority over the choices that define the final output's quality and appropriateness.

VII. FUTURE SCOPE

The current ARTURE implementation establishes a working foundation. Several directions for development have been identified that would meaningfully expand the platform's capability, reach, and cultural depth.

A. Advanced Generative Modalities and 3D Design

The most technically ambitious near-term extension is the integration of latent diffusion models and multimodal transformers into the generation pipeline. Diffusion-based image synthesis, if tightly coupled with the existing JSON rendering layer rather than used to produce final raster outputs, could expand the range of background imagery and decorative elements available within the editable canvas. Extending the platform to support three-dimensional product mockups and animated graphic sequences — accessible through the same natural language interface as the current 2D tools — would substantially broaden its commercial relevance [3], [4].

B. Real-Time Collaboration and Mobile Expansion

A significant planned enhancement is simultaneous multi-user editing, using Operational Transformation or Conflict-free Replicated Data Type (CRDT) algorithms to synchronize canvas state across concurrent editors in real time [5]. This capability would make ARTURE viable for distributed design teams, enabling clients and designers to interact with a shared canvas simultaneously. Extending the platform to native iOS and Android applications will further remove device dependency, making professional-quality design accessible in mobile-first contexts — particularly relevant for the small business and content creator segment identified as core users in the Mumbai region.

C. Adaptive Recommendation and Personalization

As the platform accumulates usage data, an adaptive recommendation engine would allow it to move from reactive to proactive assistance — surfacing relevant color schemes, typographic pairings, and layout alternatives before a user explicitly requests them, based on learned patterns from the user's design history and brand preferences. Privacy-preserving federated learning approaches would enable personalization without centralized storage of sensitive user design content.

D. Cultural Range Expansion Through Community Feedback

The current cultural template repository covers the Indian subcontinent with meaningful depth, but ARTURE's architectural approach is equally applicable to other regional contexts. Structured community feedback mechanisms are planned that would allow local designers and cultural practitioners to contribute templates, validate cultural accuracy, and flag inappropriate outputs. These contributions would update both the template repository and the vector index, allowing the system's cultural competence to grow continuously with its user base. Expansion to Southeast Asian, East African, and Latin American design traditions has been identified as the next priority area [8], [9].

VIII. CONCLUSION

This paper has presented ARTURE, an AI-enhanced web-based design platform that addresses three fundamental limitations of current AI-assisted design tools: the absence of post-generation editability, the cultural homogeneity of template and training corpora, and the disconnection between generation and editing pipelines. By integrating NLP-driven prompt interpretation, Pinecone-based vector semantic retrieval, a hybrid AI inference layer combining cloud LLMs with browser-side WebGPU computation, and Fabric.js canvas rendering, ARTURE converts natural language design intent into fully editable, multi-layer design artefacts within a single unified interface.

The platform's culturally adaptive template repository — developed with specific attention to Indian festival aesthetics and regional visual traditions — demonstrates that culturally specific AI design assistance is technically achievable and practically valuable. Evaluation results confirm meaningful improvements in retrieval accuracy, editing accessibility, and user satisfaction relative to existing tools.

The broader implication of this work is that the most useful framing for AI in creative fields is not replacement but structured augmentation. When AI systems handle the mechanical work of layout composition, template selection, and content placement, they reduce the entry barrier to professional-quality design without diminishing the role of human creative judgment. ARTURE is built around this principle, and the results support it: designers using the platform consistently reported spending more of their time on decisions they cared about and less time on tasks that felt purely technical — a shift toward more human-centred, AI-assisted creative work that this platform is designed to advance.

IX. ACKNOWLEDGMENT

The authors wish to thank Prof. Mahendra Patil, Head of Department, Computer Engineering, Atharva College of Engineering, Mumbai, for his sustained guidance, constructive feedback, and support throughout the development of the ARTURE platform. The authors also acknowledge the faculty of the Department of Computer Engineering at Atharva College of Engineering for their suggestions during the research and implementation process. Thanks are extended to the student participants who contributed time and feedback during evaluation sessions, and to the families of all authors for their encouragement throughout the project.

REFERENCES

[1] U. Nagargoje, S. Naral, D. Palve, K. Lahare, and S. Ladge (2025), "Artificial Intelligence in Modern Graphic Design: Transforming Creativity and Workflow," International Journal of Innovative Research and Technology (IJIRT), vol. 11, no. 12. [Online]. Available: https://ijirt.org

[2] S. Sharma and J. Prakash (2024), "Generative Design: AI-Powered Creativity in Graphic Design," International Journal of Creative Research Thoughts (IJCRT). [Online]. Available: https://ijcrt.org

[3] K. Fleischmann (2024), "Generative Artificial Intelligence in Graphic Design Education: A Student Perspective," Canadian Journal of Learning and Technology (CJLT/RCAT), vol. 50, no. 1. [Online]. Available: https://doi.org/10.21432/cjlt28342

[4] P. O'Donovan, A. Agarwala, and A. Hertzmann (2015), "DesignScape: Design with Interactive Layout Suggestions," Proc. ACM CHI Conference on Human Factors in Computing Systems, Seoul, pp. 1221–1224. [Online]. Available: https://doi.org/10.1145/2702123.2702149

[5] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu (2020), "Attribute-Conditioned Layout GAN for Automatic Graphic Design," IEEE Trans. Visualization and Computer Graphics, vol. 27, no. 10, pp. 4039–4048. [Online]. Available: https://doi.org/10.1109/TVCG.2020.2999335

[6] Y. Huang, T. Lv, L. Cui, Y. Lu, and F. Wei (2022), "LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking," Proc. 30th ACM Int. Conf. on Multimedia (MM'22), Lisbon, pp. 4083–4091. [Online]. Available: https://doi.org/10.1145/3503161.3548112

[7] M. Hui, Z. Zhang, X. Zhang, W. Xie, Y. Wang, and Y. Lu (2023), "Unifying Layout Generation with a Decoupled Diffusion Model," Proc. IEEE/CVF CVPR, Vancouver, pp. 1942–1951. [Online]. Available: https://doi.org/10.1109/CVPR52729.2023.00193

[8] Z. Tang, C. Wu, J. Li, and N. Duan (2023), "LayoutNUWA: Revealing the Hidden Layout Expertise of Large Language Models," arXiv preprint arXiv:2309.09506. [Online]. Available: https://arxiv.org/abs/2309.09506

[9] Z. Zhang et al. (2025), "CreatiPoster: Towards Editable and Controllable Multi-Layer Graphic Design Generation," ACM Transactions on Graphics. [Online]. Available: https://doi.org/10.1145/3687972

[10] O. Peckham et al. (2025), "Artificial Intelligence in Generative Design: A Structured Review of Trends and Opportunities," Designs, vol. 9, no. 79. [Online]. Available: https://doi.org/10.3390/designs9040079

[11] M. Feller, Y. Xu, L. Skitka, and S. Lerman (2023), "Mitigating Bias in Algorithmic Design: A Multifaceted Approach," Proc. ACM FAccT, Chicago, pp. 812–824. [Online]. Available: https://doi.org/10.1145/3593013.3594066

[12] J. McCormack, T. Gifford, and P. Hutchings (2019), "Autonomy, Authenticity, Authorship and Intention in Computer Generated Art," Proc. International Conference on Computational Creativity (ICCC), Charlotte, NC. [Online]. Available: https://computationalcreativity.net/iccc2019

[13] K. Wong, B. Friedman, M. Sundararajan, M. Resnick, and J. Keller (2023), "The Trouble with Bias in Algorithmic Design Tools," Proc. ACM CHI, Hamburg. [Online]. Available: https://doi.org/10.1145/3544548.3580701

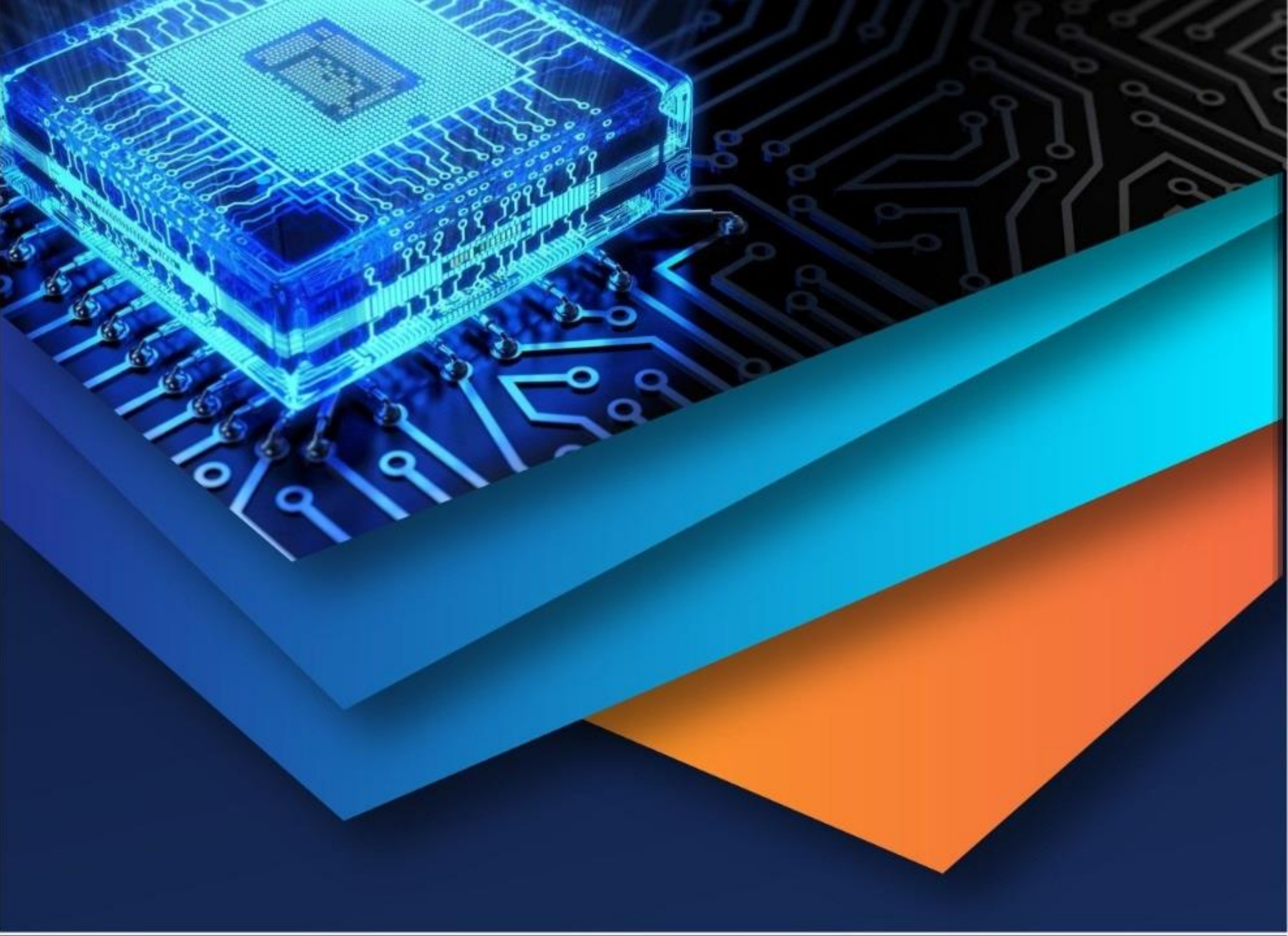
[14] M. Kretzschmar et al. (2024), "Evaluating the Role of Generative AI in Product Development and Design — A Systematic Review," Proc. NordDesign 2024, Odense, Denmark. [Online]. Available: https://doi.org/10.35199/NORDDESIGN2024

[15] P. M. Khanolkar, A. Vrolijk, and A. Olechowski (2023), "A Case Study of Decision Making Behind the Automation of a Composites Based Design Process," Proc. Design Society, Bordeaux, France. [Online]. Available: https://doi.org/10.1017/pds.2023.180

[16] Pinecone Systems Inc. (2024), "Pinecone Vector Database Documentation," Technical Documentation. [Online]. Available: https://docs.pinecone.io

[17] Fabric.js Contributors (2024), "Fabric.js — HTML5 Canvas Library," Open Source Documentation. [Online]. Available: http://fabricjs.com/docs

[18] Vercel Inc. (2024), "Next.js Documentation," Developer Documentation. [Online]. Available: https://nextjs.org/docs



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY