International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

PromptPixel-Automatic Prompt Optimization and Image Generation

Kishor Pathak
Information Technology
Vishwakarma Institute of Technology
Pune, India

Vishwatej Shende
Information Technology
Vishwakarma Institute of Technology
Pune, India

Rishi Shende
Information Technology
Vishwakarma Institute of Technology
Pune, India

Anish Shete
Information Technology
Vishwakarma Institute of Technology
Pune,India

Arya Shinde
Information Technology
Vishwakarma Institute of Technology
Pune, India

Rutuja Solanke
Information Technology
Vishwakarma Institute of Technology
Pune, India

Abstract **- PromptPixel is an interactive text-to-image system that helps users refine prompts without losing their original intent. Many prompt-enhancement tools rewrite user input, which can change its meaning or omit important details. PromptPixel preserves the original wording and checks the prompt against five visual dimensions: style, composition, lighting, mood and color, and detail. For dimensions it does not detect, it suggests additions and explains their purpose. A before-and-after coverage indicator summarizes the detected dimensions, and users can accept, edit, or remove suggestions before generating an image. The system also lets users revisit prompts and provide feedback on generated images. Feedback can be retrieved for similar prompts to inform later suggestions. PromptPixel combines prompt analysis, explainable recommendations, image generation, and feedback-based refinement in one workflow. Its rule-based optimizer is intended to make prompt changes visible and editable; its effect on image quality and user outcomes requires further evaluation.**

Keywords - ***Text-to-image generation, prompt optimization, explainable AI, user feedback, interactive systems.***

I. INTRODUCTION

Text-to-image generation systems turn natural-language descriptions into images. Advances in latent diffusion and other text-conditioned models have made this possible [9], [10]. However, getting an image that matches a user's intention can still be difficult. Users may know what they want to depict but not know how to describe visual details such as composition, lighting, style, or color. Prompt writing can therefore become a barrier between a creative idea and a useful result. Research on prompt collections and prompt-engineering practices has also noted the variety of prompts people use and the trial-and-error involved in finding effective descriptions [2], [4].

Researchers have explored different ways to address this challenge. Promptist transforms user input into prompts that suit a text-to-image model [1], while discrete optimization methods aim to discover interpretable prompts [3]. Other approaches use language-model feedback and iterative search to improve prompts [6], [8]. Image-generation research has also explored prompt editing and ways to control generated content [5], [11], [12]. These methods show the potential of prompt optimization and image control. However, users may not always be shown which parts of their original prompt were preserved, what was added, or why a change was suggested. Human–AI interaction guidelines emphasize that systems should communicate their capabilities and help users understand and correct system behavior [7].

This paper presents PromptPixel, an integrated text-to-image workflow that combines prompt assessment, optimization, image generation, and user-guided refinement. Its main contribution is an intent-locked, explainable visual-slot optimization method. The method preserves the user's original prompt and checks it for five visual dimensions: style, composition, lighting, mood and color, and detail. When the system does not detect a dimension, it suggests an addition and briefly explains its purpose. A coverage indicator shows which dimensions were detected before and after optimization. Users can edit or remove suggestions before generating an image.

After generation, users can return to the prompt they used and refine it rather than starting again. They can also describe what did not meet their expectations. This feedback can be retrieved for similar prompts and inform future suggestions. PromptPixel brings explanation, editing, generation, and feedback together in one interface. Its contribution is this transparent, integrated workflow, not a new image-generation model or a guarantee of better images. Since the optimizer uses rule-based dimension detection and templates, its effectiveness in improving image alignment, usability, and prompt-writing skills remains to be evaluated.

II. LITERATURE REVIEW

Text-to-image generation allows users to create images from written descriptions, but the result depends in part on how clearly the prompt expresses the user's idea. Rombach *et al.* [9] introduced latent diffusion models,

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

which provide a technical foundation for many image-generation systems. Ramesh *et al.* [10] presented a text-conditional image-generation framework that connects text and visual representations. These studies show how written descriptions guide image generation. Wang *et al.* [4] introduced DiffusionDB, a large collection of prompts and generated images that helps researchers study how people describe their ideas. Liu and Chilton [2] examined prompt-writing practices and provided guidelines to help users create effective prompts. Together, these works highlight the challenges users face when trying to express visual ideas clearly.

Several studies have explored automatic prompt improvement. Hao *et al.* [1] proposed Promptist, which rewrites user prompts to better suit a text-to-image model. Wen *et al.* [3] introduced PEZ, a gradient-based approach for finding readable prompts. Zhou *et al.* [6] proposed Automatic Prompt Engineer, which generates prompt candidates and selects them based on task performance. Pryzant *et al.* [8] introduced ProTeGi, which uses natural-language feedback and search to improve prompts. These works demonstrate the potential of automated optimization. However, they do not focus on showing users each visual change and its reason as part of an interactive image-generation workflow.

Other research has explored ways to give users more control over generated images. Hertz *et al.* [5] introduced Prompt-to-Prompt, which uses attention control to edit images by changing prompt words. Gal *et al.* [11] proposed textual inversion, which learns a new text representation for a visual concept using example images. Zhang *et al.* [12] introduced ControlNet, which guides image generation using additional spatial information such as edges or pose. These methods provide ways to edit or control images, but they do not primarily focus on helping users understand and improve a written prompt.

Amershi *et al.* [7] presented guidelines for human–AI interaction, including making system behavior understandable and allowing users to correct or guide AI outputs. These principles are relevant to prompt-enhancement tools, where users should be able to review and edit suggestions rather than accept an unexplained rewrite.

Overall, previous research has contributed to image generation [9], [10], prompt analysis [2], [4], prompt optimization [1], [3], [6], [8], image control [5], [11], [12], and human–AI interaction [7]. However, there remains a need for a system that improves prompts while preserving the user's original wording and explaining its suggestions. PromptPixel addresses this need by checking five visual dimensions: style, composition, lighting, mood and color, and detail. It identifies dimensions that may be missing, suggests relevant additions, and explains each suggestion. Users can edit the enhanced prompt before generating an image. The system emphasizes transparency and user involvement; its effectiveness in improving image alignment, usability, and prompt-writing skills remains to be evaluated.

III. METHODOLOGY

PromptPixel follows a design-and-implementation approach. Users may know what image they want but find it difficult to describe visual details, while automatic rewriting can make it hard to see what changed. The proposed method is called intent-locked visual-slot optimization. "Intent-locked" means that the user's original prompt is preserved at the beginning of the enhanced prompt. A "visual slot" is a category of image detail, such as style or lighting. Here, "optimization" means checking for these categories and suggesting additions when the system does not detect them; it does not mean that a trained model calculates an optimal prompt or guarantees a better image. This approach draws on prompt-engineering guidance to describe a subject with relevant style and visual context [2], as well as human-AI interaction principles that allow users to understand and correct AI-assisted output [7]. PromptPixel does not silently replace the prompt; it presents suggestions with explanations that users can edit or remove before generating an image. The prototype uses rule-based phrase matching. A dimension is marked as present when the prompt contains a word or phrase from that dimension's pattern list. If no matching phrase is found, the system appends a general suggestion for that dimension. Table I summarizes the visual dimensions, example detection cues, and example suggestions.

TABLE I. VISUAL DIMENSIONS USED IN PROMPT OPTIMIZATION

Dimension	What it describes	Example detection cues	Example suggestion
Style	The visual medium or appearance	cinematic, watercolor, anime	A coherent visual style that complements the subject
Composition	Subject placement and framing	close-up, wide shot, foreground	A clear focal subject with balanced framing
Lighting	How light and shadow shape the scene	golden hour, backlit, neon	Soft, directional lighting suited to the scene
Mood and color	Feeling and color treatment	serene, dramatic, vibrant, palette	A cohesive color palette and atmosphere
Detail	Texture and visual clarity	intricate, texture,	Crisp, intentional details with

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

		sharp focus	clearly separated forms

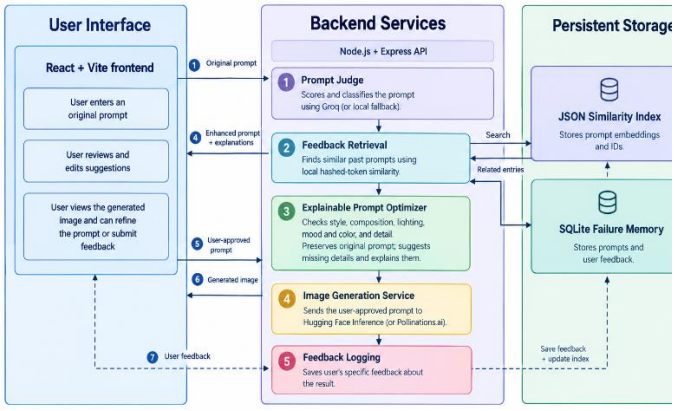


Fig. 1 Architecture Diagram of Prompt Pixel

The frontend is built with React and provides the prompt-entry, enhancement, result, and feedback screens. The backend uses Node.js and Express to handle prompt assessment and optimization, run failure-memory searches, and process image-generation requests. The optimization module checks the user's text against five rule-based visual dimensions. It creates an enhanced prompt and provides a structured explanation for each suggestion. When a Groq API key is configured, the prompt judge uses the Groq service. If no key is configured, it uses a local rule-based fallback. SQLite stores feedback about unsatisfactory results, while a local hashed-token vector index retrieves past attempts with sufficiently similar prompt wording. Image generation uses an external provider, so its availability and account limits can affect whether an image is returned.

TABLE II. MAIN SYSTEM COMPONENTS

Component	Role	Output
Prompt judge	Scores and classifies the initial prompt	Type, score, and feedback
Feedback retrieval	Finds earlier attempts with similar wording	Related feedback entries
Visual-slot optimizer	Detects missing dimensions and creates suggestions	Enhanced prompt, reasons, and coverage
Image generator	Sends the final edited prompt to a configured provider	Image data and image format
Feedback storage	Saves the user's stated issue	Entry for later retrieval

Table II summarizes the five main components of PromptPixel and their respective functions. Each component handles a specific task, from prompt evaluation and feedback retrieval to prompt enhancement, image generation, and feedback storage. Together, they support an explainable and structured prompt optimization workflow.

The workflow starts when the user enters a natural-language description. The backend first assesses and classifies the prompt. It then searches the feedback memory for similar earlier prompts. Similarity is calculated using local hashed-token vectors, so it is an approximate lexical match rather than a deep semantic understanding of the prompt. The optimizer checks which visual dimensions are missing and adds suggestions for those dimensions. Each suggestion includes a brief reason, along with a coverage summary showing which dimensions were detected before and after optimization. If retrieved feedback includes a specific, non-generic reason, the system can include it as a correction. The user can then review and edit the enhanced prompt before requesting image generation. Once the image is returned, the user can download it, refine the prompt, or submit feedback explaining what did not meet their expectations. The feedback is stored and may be retrieved during a later attempt with a similar prompt.

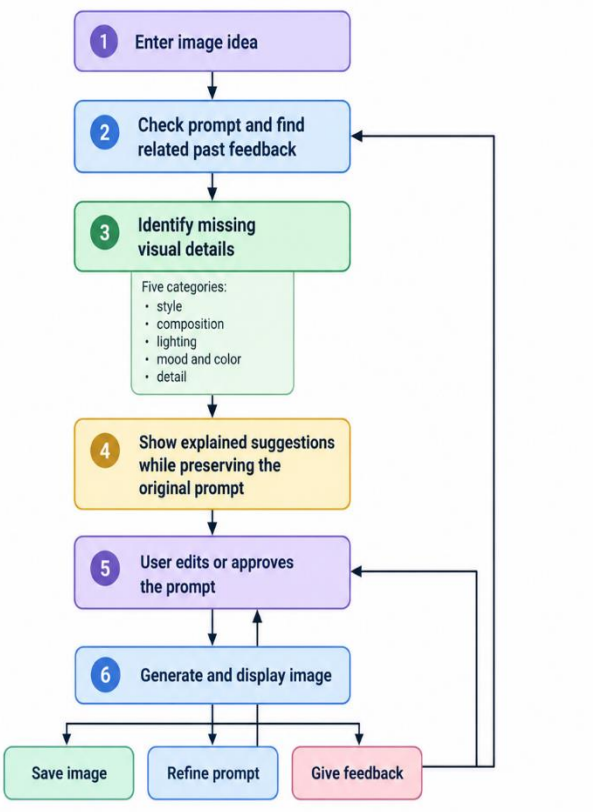


Fig. 2. PromptPixel workflow

The workflow is organized as separate functions so that assessment, optimization, and image generation can operate independently. The optimizer preserves the original prompt and adds suggestions rather than rewriting its subject. This allows users to compare the original and enhanced versions and reject changes that do not match their intention. The

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

implementation returns the suggested additions as structured data, allowing the interface to show both the revised prompt and the reason for each change. The user can modify the final prompt before generation, so the optimizer provides assistance rather than imposing a final result.

IV. RESULTS AND DISCUSSION

PromptPixel was tested as a software prototype. The checks covered the prompt-optimization logic, backend responses, and frontend production build. The tests confirm that the system can preserve an input prompt, identify missing visual dimensions, explain its suggestions, and return the optimization result to the interface. They do not establish that generated images are better or that users learn prompt-writing skills more effectively.

TABLE III. IMPLEMENTATION TEST RESULTS

Test	Result	What it confirms
Prompt optimizer unit tests	3 passed, 0 failed	Original wording is retained, already specified dimensions are not redundantly added, and specific feedback can be included
Frontend production build	Passed	The React interface builds successfully
Backend syntax checks	Passed	The updated backend files parse successfully
API health and enhancement smoke tests	Passed	The backend responds and returns enhanced prompts with explanations
Image-generation provider test	Not completed successfully	The provider returned an access/credits error, so successful image generation could not be confirmed in this environment

The optimizer uses a five-dimension coverage indicator for style, composition, lighting, mood and color, and detail. When a prompt does not cover a dimension, the system adds a suggestion intended to address it. This indicator helps describe the optimizer's behavior, but it is not an independent measure of image quality.

TABLE IV. RESULTS THAT CAN BE REPORTED FROM CURRENT TESTING

Measure	Current finding	Interpretation
Optimizer tests passed	3/3	Core optimizer behavior passed the included tests
Frontend production build	Passed	The application can be built for production
API health/enhancement checks	Passed	Basic backend routes responded during smoke testing
Successful image-generation requests confirmed	0 in this environment	Provider access or credits prevented verification; this is not a measured failure rate

The coverage score is not an independent measure of image quality because the same five dimensions are used to generate suggestions and calculate the score. Therefore, a higher score after optimization is expected by design and cannot be used as evidence that the generated images improved. The results show that PromptPixel's main workflow is implemented and that the prompt-optimization component passed the targeted tests.-

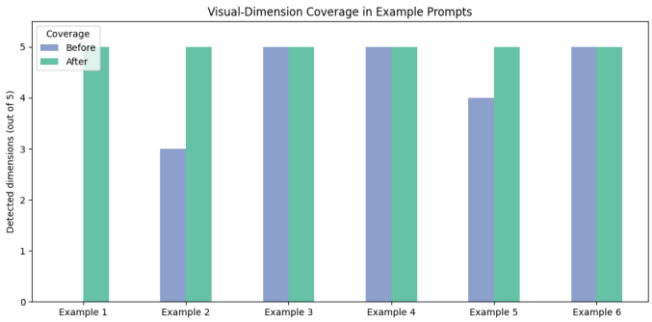


Fig. 3. Visual-dimension coverage before and after enhancement.

Fig. 3 shows visual-dimension coverage before and after prompt enhancement for six illustrative example prompts. Coverage increases for Examples 1, 2, and 5, where dimensions were missing and suggestions were added. Examples 3, 4, and 6 already cover all five dimensions, so no additional suggestions are needed. These examples illustrate how the optimizer's coverage indicator responds to the presence or absence of the specified dimensions; they do not evaluate generated-image quality.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

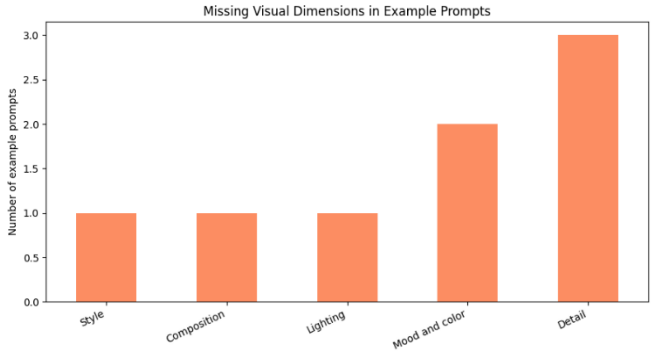


Fig. 4. Missing visual dimensions detected in six illustrative prompts.

Fig. 4 shows the missing visual dimensions detected in the six illustrative prompts. Detail is the most frequently missing dimension, appearing in three prompts, followed by mood and color, which are missing in two prompts. Style, composition, and lighting are each missing in one prompt. These counts describe the selected examples only. They should not be generalized to prompts overall or interpreted as evidence that adding dimensions improves image-generation quality.

V. CONCLUSION

This paper presented PromptPixel, an integrated system for text-to-image generation that aims to make prompt optimization more transparent and accessible. Its central method, intent-locked visual-slot optimization, preserves the user's original wording and checks for five visual dimensions: style, composition, lighting, mood and color, and detail. When a dimension is not detected, the system suggests an addition and explains its purpose. Users can edit or remove these suggestions before image generation. PromptPixel brings prompt assessment, image generation, and post-generation refinement into one workflow. Users can return to the prompt they originally used rather than starting over, and feedback about what did not work can inform later suggestions for similar prompts. This combination is the project's main contribution: an editable, feedback-aware prompt-optimization experience that explains its changes rather than presenting a rewritten prompt as a black box. The current implementation is a working prototype, not evidence that the method improves image quality or user outcomes. The optimizer detects visual dimensions with rules and templates, while feedback retrieval relies on prompt similarity. Its effectiveness must therefore be established through formal evaluation. Future studies can examine whether the system helps users express visual intent, understand prompt construction, and refine generated images more efficiently.

VI. FUTURE SCOPE

Future work should evaluate PromptPixel with users and compare it with a workflow that does not provide prompt optimization. A study could measure task-completion time, the number of prompt revisions, users' understanding of suggested changes, perceived control, and satisfaction. Generated images could also be assessed by users for alignment with their stated intent, with image-text similarity measures used as supplementary rather than definitive evidence. Such an evaluation would help determine whether explainable suggestions improve the creative process and which user groups benefit most. The optimizer could be improved by replacing rule-based phrase matching with semantic detection that recognizes visual concepts expressed in varied language. For example, it should recognize that "a close view" may indicate framing and that "soft evening sun" describes lighting. Future versions could let users choose which dimensions to optimize, set preferences, and specify constraints such as aspect ratio or elements to avoid. Model-aware optimization could adapt suggestions to the image-generation provider while continuing to show users what changed and why. Feedback retrieval could also be improved beyond basic prompt similarity, and users could be given more control over stored feedback. Further work should examine privacy, accessibility, and reliability across different users and prompt styles.

REFERENCES

[1] Y. Hao, Z. Chi, L. Dong, and F. Wei, "Optimizing prompts for text-to-image generation," in *Advances in Neural Information Processing Systems*, vol. 36, 2023. [Online]. Available: https://arxiv.org/abs/2212.09611

[2] V. Liu and L. B. Chilton, "Design guidelines for prompt engineering text-to-image generative models," in *Proc. CHI Conf. Human Factors Comput. Syst. (CHI '22)*, 2022, Art. no. 384, pp. 1–23, doi: 10.1145/3491102.3501825.

[3] Y. Wen, N. Jain, J. Kirchenbauer, M. Goldblum, J. Geiping, and T. Goldstein, "Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery," *arXiv preprint arXiv:2302.03668*, 2023. [Online]. Available: https://arxiv.org/abs/2302.03668

[4] Z. J. Wang, E. Montoya, D. Munechika, H. Yang, B. Hoover, and D. H. Chau, "DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models," *arXiv preprint arXiv:2210.14896*, 2022. [Online]. Available: https://arxiv.org/abs/2210.14896

[5] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, "Prompt-to-prompt image editing with cross attention control," *arXiv preprint arXiv:2208.01626*, 2022. [Online]. Available: https://arxiv.org/abs/2208.01626

[6] Y. Zhou *et al.*, "Large language models are human-level prompt engineers," *arXiv preprint arXiv:2211.01910*, 2022. [Online]. Available: https://arxiv.org/abs/2211.01910

[7] S. Amershi *et al.*, "Guidelines for human-AI interaction," in *Proc. CHI Conf. Human Factors Comput. Syst. (CHI '19)*, 2019, Art. no. 3, pp. 1–13, doi: 10.1145/3290605.3300233.

[8] R. Pryzant, D. Iter, J. Li, Y. T. Lee, C. Zhu, and M. Zeng, "Automatic prompt optimization with 'gradient descent' and beam search," *arXiv preprint arXiv:2305.03495*, 2023. [Online]. Available: https://arxiv.org/abs/2305.03495

[9] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 10684–10695, doi: 10.1109/CVPR52688.2022.01042.

[10] A. Ramesh *et al.*, "Hierarchical text-conditional image generation with CLIP latents," *arXiv preprint arXiv:2204.06125*, 2022. [Online]. Available: https://arxiv.org/abs/2204.06125

[11] R. Gal *et al.*, "An image is worth one word: Personalizing text-to-image generation using textual inversion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 192–202.

[12] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 3836–3847.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)