

RANANO-Recreating Nano Banana Using Diffusion Model

Manjushree K R¹, Sakshi S², Sanjana Prakash Jadamali³, Shreya Udaya Vernekar⁴, Vinuthana Guttal⁵

^{1,2,3,4,5} Information Science and Engineering, Bapuji Institute of Engineering and Technology, Davangere, Karnataka, India.

To Cite this Article: Manjushree K R¹, Sakshi S², Sanjana Prakash Jadamali³, Shreya Udaya Vernekar⁴, Vinuthana Guttal⁵, "RANANO-Recreating Nano Banana Using Diffusion Model", *Indian Journal of Computer Science and Technology*, Volume 05, Issue 03 (September-December 2026), PP: 365-370.



Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: RANANO is an integrated artificial intelligence-based multimedia creation platform developed to combine multiple creative functionalities within a single web-based interface. The system addresses the fragmented workflow involved in using separate applications for image generation, image enhancement, video creation, animated text generation, and user guidance. RANANO uses a trained diffusion model for text-to-image generation, where the trained model checkpoint is integrated into the application for inference. The platform also incorporates image enhancement using super-resolution, video generation from selected images, animated text generation, and a voice assistant for interactive guidance. The system follows a modular architecture in which the frontend communicates with backend services responsible for model inference, image processing, multimedia generation, and output management. The diffusion model is trained using a prepared dataset, while the trained checkpoint is subsequently loaded during application execution without repeated training. Functional testing was conducted across the major modules, including checkpoint loading, prompt processing, image generation, image enhancement, video generation, animated text, voice interaction, output display and storage, and frontend-backend integration. All twelve test cases documented in the project were successfully passed. The results demonstrate that RANANO provides a unified and modular approach for AI-assisted multimedia creation while reducing the need to switch between multiple tools. The system also provides scope for future improvements in computational efficiency, processing time, and multimedia generation capabilities.

Key Word: Artificial intelligence, diffusion models, image generation, image enhancement, multimedia creation, super-resolution.

I. INTRODUCTION

The rapid advancement of Artificial Intelligence (AI), particularly Generative AI, has significantly influenced the field of digital content creation. AI-based techniques are increasingly being used to generate and process visual content, allowing users to create images and other multimedia outputs with reduced manual effort. Diffusion models have emerged as an important deep learning approach for image generation, in which a model learns to generate meaningful visual content from noise. However, multimedia creation commonly involves multiple independent tools for tasks such as image generation, image enhancement, video creation, and animation, resulting in a fragmented and time-consuming workflow.

Existing multimedia content creation systems generally provide individual functionalities through separate applications. Image-generation systems can produce images from textual prompts, while image-enhancement tools are used for restoration and quality improvement. Similarly, video-generation applications can combine images into video sequences, but users may need to switch between different platforms to complete a complete creative task. The project report identifies additional challenges including high computational requirements, complex user interfaces, limited interactive guidance, and the need for multiple tools to accomplish different multimedia operations.

To address these limitations, this work presents RANANO: Recreating Nano Banana Using Diffusion Model, an integrated AI-based multimedia creation platform developed as a web-based application. The proposed system combines multiple functionalities within a single interface, including text-to-image generation using a trained diffusion model, image enhancement using super-resolution, video generation from selected images, animated text generation, and a voice assistant for user guidance.

The image-generation component uses a trained diffusion-model checkpoint that is integrated into the RANANO application for inference. The model is trained separately using a prepared dataset, and the resulting checkpoint is subsequently loaded during application execution. This separates the training stage from normal user interaction and allows the application to use the trained model without repeated training. The generated images can then be processed by the image-enhancement, video-generation, and other multimedia modules according to the user's requirements.

The main contribution of this work is the integration of multiple AI-assisted multimedia functionalities into a single modular platform, rather than developing each functionality as an isolated application. The developed system is evaluated through functional testing covering model checkpoint loading, prompt processing, image generation, image enhancement, video generation, animated text, voice interaction, output handling, and frontend-backend communication.

II. MATERIAL AND METHODS

RANANO-Recreating Nano Banana Using Diffusion Model was developed as a web-based artificial intelligence platform for multimedia content creation. The system integrates text-to-image generation, image enhancement, video generation, animated text generation, and voice-assisted interaction within a unified interface. The methodology involved dataset preparation, diffusion-model training, model integration, multimedia processing, and functional testing of the implemented modules.

Study Design: Development and functional evaluation of an AI-based multimedia application.

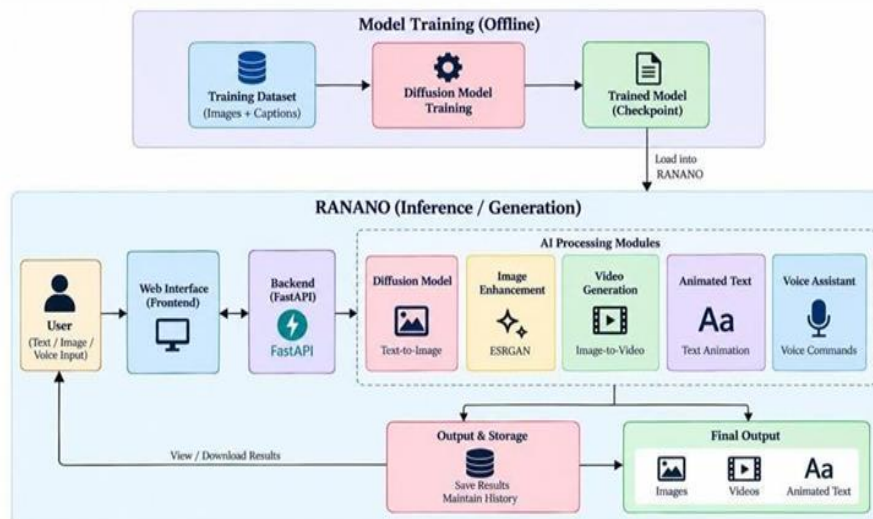
Study Location: The project was designed and implemented in the Department of Information Science and Engineering, Bapuji Institute of Engineering and Technology, Davangere, Karnataka, India.

Study Duration: The project was carried out during the academic year 2026–2027. Exact development and evaluation dates were not specified.

System Components: The application consists of a web-based frontend, backend API services, a trained diffusion-model checkpoint, an image-enhancement module, a video-generation module, an animated-text module, a voice assistant, and output-handling functionality.

Software and Hardware Requirements: The software environment includes Python 3.8 or later, PyTorch, NumPy, Transformers, OpenCV, FastAPI, and the Web Speech API. The development environment uses Windows 10/11 and Visual Studio Code. The project specifies an Intel Core i7 or equivalent processor and 16 GB RAM as a recommended hardware configuration. Actual computational requirements depend on the model and execution environment.

RANANO follows a modular architecture consisting of a web-based frontend, FastAPI backend services, a trained diffusion model, image-enhancement and multimedia-processing modules, and output storage. The trained model checkpoint is loaded for inference, while user requests are routed to the corresponding module. The processed results are returned to the interface for display or download.



Dataset Preparation and Model Training: The image-generation component uses a prepared dataset containing image data and associated caption information. The dataset is preprocessed into a format suitable for diffusion-model training. During training, noise is introduced into image representations, and the model learns to estimate and progressively remove the noise to reconstruct meaningful visual content.

After training, the model parameters are saved as a checkpoint. This checkpoint is subsequently integrated into the RANANO application and loaded during inference. Consequently, the model does not need to be retrained for each image-generation request. The dataset size, source, number of training epochs, batch size, learning rate, and optimizer configuration were not specified in the available project documentation.

Text-to-Image Generation: The text-to-image module accepts a textual prompt through the web interface. The backend validates the prompt and processes the input before initiating image generation. Empty prompts are rejected, while valid prompts are passed to the generation pipeline.

The generation process begins with a random noise representation. The trained diffusion model iteratively estimates the noise at successive sampling steps, gradually transforming the initial representation into an image. The resulting image is returned to the frontend for display and can subsequently be used by other multimedia modules. The interaction diagram between the frontend, backend, diffusion model, enhancement module, and storage component is illustrated in the sequence diagram.

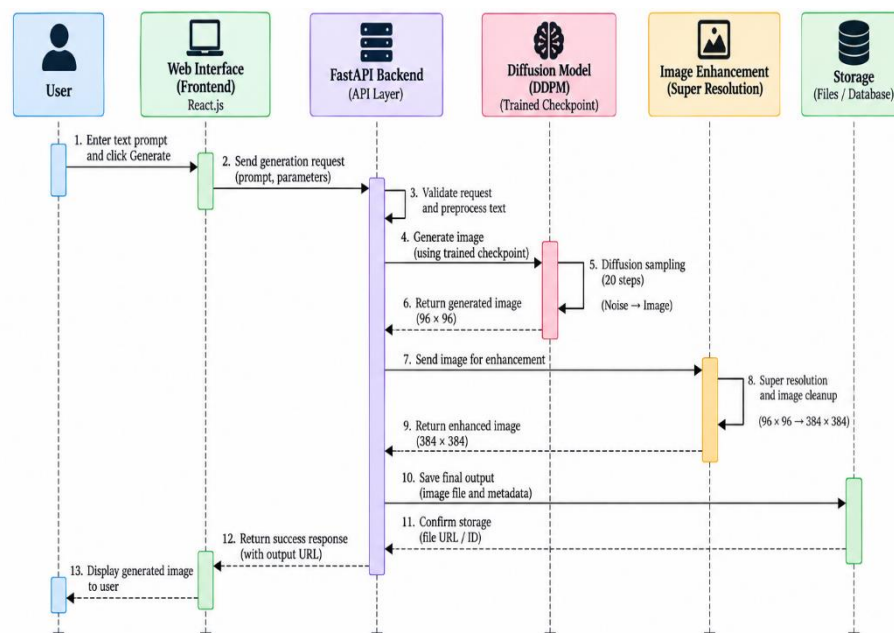


Image Enhancement: The image-enhancement module processes generated or uploaded images to improve their resolution and visual clarity. The implementation uses Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) with a reported $\times 4$ upscaling configuration. The input image is processed by the enhancement module, and the resulting image is returned to the application interface.

The enhanced output can be viewed by the user and used for subsequent multimedia operations. Quantitative image-quality metrics, including Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), were not documented in the available results.

Video Generation and Animated Text: The video-generation module accepts multiple images selected by the user. The images are arranged in the specified sequence and treated as individual frames. These frames are combined to generate a video file, which is made available through the web interface for viewing and storage.

The animated-text module accepts user-provided text and a selected animation style. The system processes the text according to the selected option and generates the corresponding animated textual content. These features enable users to create multimedia outputs within the same application without switching between separate tools.

Voice Assistant and System Integration: The voice assistant provides an additional method of interacting with the platform through voice commands. Speech input is processed into a form that can be used by the application, enabling users to navigate supported functionalities and receive operational guidance. The Web Speech API is identified as part of the software environment for voice interaction.

The system follows a modular architecture in which the frontend communicates with backend services through API requests. Based on the selected functionality, the backend routes each request to the corresponding processing module. The processed result is returned to the frontend for display or storage. This architecture supports the integration of the individual modules within a common interface.

Statistical Analysis: The evaluation was based on functional testing of the implemented software modules. The available project documentation does not report a statistical sample, experimental control group, quantitative performance dataset, or inferential statistical analysis. Therefore, statistical significance testing was not applicable to the reported functional test results. Future evaluation could include repeated generation trials, image-quality measurements, execution-time analysis, and comparative testing against relevant baseline methods.

III. RESULT

The RANANO system was implemented as an integrated AI-based multimedia platform combining diffusion-model-based image generation, image enhancement, video generation, animated-text generation, and voice-assisted interaction. The system was evaluated through functional testing of individual modules and their integration with the web interface. The implementation involved two major stages: training the diffusion model using the prepared dataset and integrating the trained model checkpoint into RANANO for inference and multimedia generation.

3.1 Diffusion Model Training and Integration

The prepared dataset was processed into a suitable format for diffusion-model training. The preprocessing stage organized the image data for use by the model. The diffusion model was subsequently trained, and the resulting checkpoint was integrated into the RANANO application.

```

import os, sys
sys.path.insert(0, PROJECT_DIR) # needed for direct imports inside the
from data.fruits360_loader import DEFAULT_CLASSES

train_dir = f"{PROJECT_DIR}/data/fruits360/Training"
print(f"{'class':<20} {'image count':>12}")
print("-" * 33)
for cls in DEFAULT_CLASSES:
    cls_dir = os.path.join(train_dir, cls)
    if os.path.isdir(cls_dir):
        n = len([f for f in os.listdir(cls_dir) if f.lower().endswith(
            print(f"{'cls':<20} {'n':>12}")
    else:
        print(f"{'cls':<20} {'MISSING':>12}")

*** Class          Image count
-----
Banana 1           490
Apple Red 1        492
Orange 1           479
Strawberry 1       492
Pineapple 1        490
Lemon 1            492
Mango 1            490
Kiwi 1             466

python train_diffusion.py --epochs 200 --image_size 96

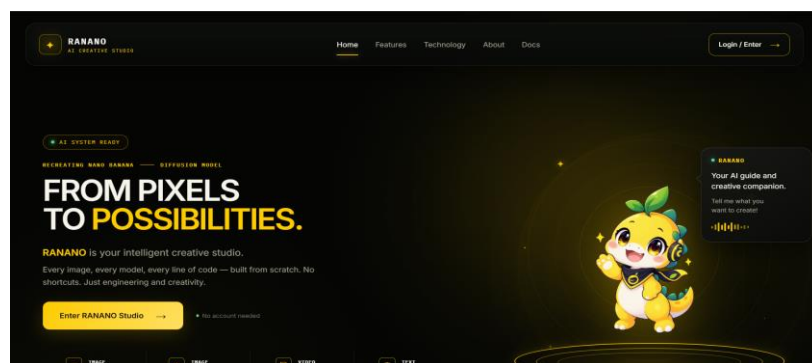
Epoch 1/200 - loss: 0.1495
Epoch 2/200 - loss: 0.0267
Epoch 3/200 - loss: 0.0195
Epoch 4/200 - loss: 0.0170
Epoch 5/200 - loss: 0.0164
Epoch 6/200 - loss: 0.0135
Epoch 7/200 - loss: 0.0118
Epoch 8/200 - loss: 0.0120
Epoch 9/200 - loss: 0.0123
Epoch 10/200 - loss: 0.0127
Saved checkpoint: checkpoints/diffusion_epoch_10.pt
Epoch 11/200 - loss: 0.0123
Epoch 12/200 - loss: 0.0123
Epoch 13/200 - loss: 0.0106
Epoch 14/200 - loss: 0.0114
Epoch 15/200 - loss: 0.0117
Epoch 16/200 - loss: 0.0105
Epoch 17/200 - loss: 0.0115
Epoch 18/200 - loss: 0.0102
Epoch 19/200 - loss: 0.0112
Epoch 20/200 - loss: 0.0103
Saved checkpoint: checkpoints/diffusion_epoch_20.pt
Epoch 21/200 - loss: 0.0100
Epoch 22/200 - loss: 0.0110
Epoch 23/200 - loss: 0.0102
Epoch 24/200 - loss: 0.0103
Epoch 25/200 - loss: 0.0092
Epoch 26/200 - loss: 0.0098
Epoch 27/200 - loss: 0.0091
Epoch 28/200 - loss: 0.0090

*** Found 20 checkpoints with preview grids

```

During normal application execution, the trained checkpoint is loaded for inference rather than retraining the model for every request. The training outputs show the model-training process, checkpoint saving, and generated sample previews at different training stages.

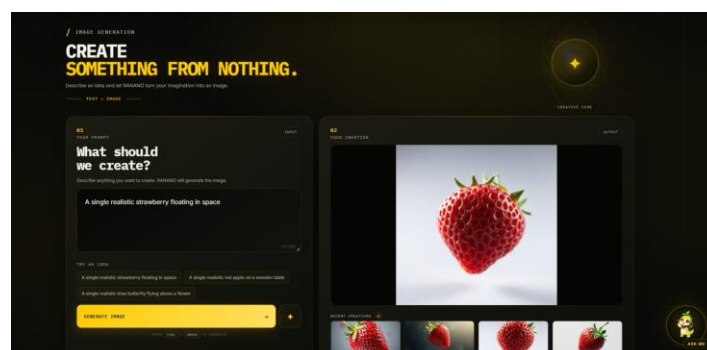
RANANO Model Integration: This snapshot shows the integration of the trained diffusion-model checkpoint into the RANANO project. During application execution, RANANO loads the previously trained checkpoint instead of training the model again.



3.2 Text-to-Image Generation

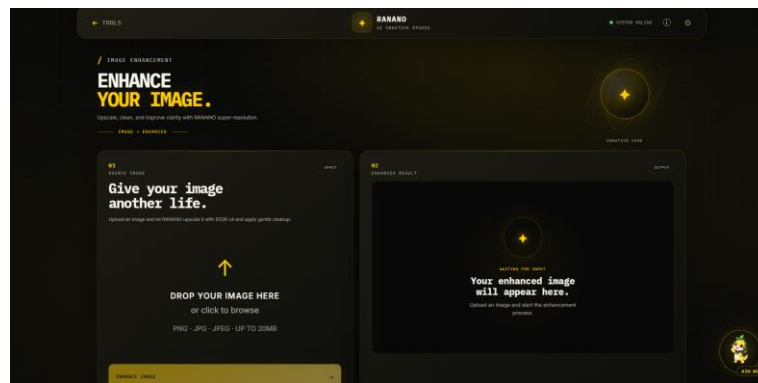
The text-to-image module accepts valid textual prompts and uses the trained diffusion model to generate corresponding images. Functional testing confirmed that valid prompts were accepted and that image generation operated successfully through the integrated model.

This functionality represents the primary generative capability of RANANO and provides visual content that can be used in other supported multimedia operations.



3.3 Image Enhancement

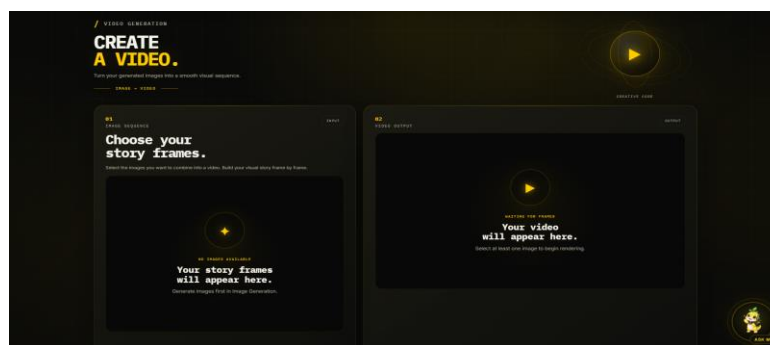
The image-enhancement module processes generated or uploaded images to improve their resolution and visual appearance. The implementation incorporates image-enhancement and super-resolution functionality. Functional testing confirmed that the module produced an enhanced output.



3.4 Video Generation

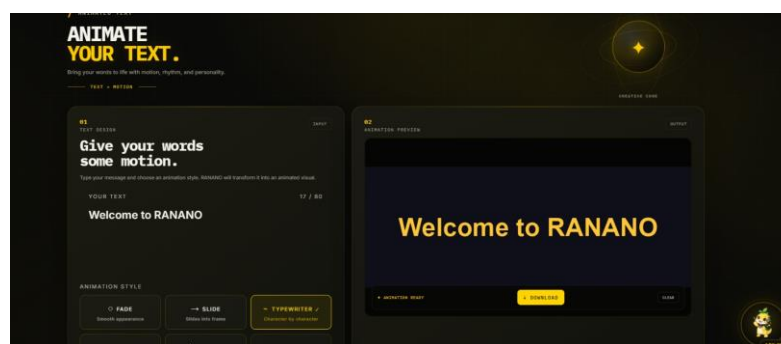
The video-generation module enables users to combine multiple images into a video sequence. The selected images are processed in sequence to produce a video output that can be displayed and stored through the application.

Functional testing confirmed that the video-generation functionality operated successfully.



3.5 Animated-Text Generation

The animated-text module accepts user-provided text and a selected animation style. The system processes the input and generates the corresponding animated text content. Functional testing confirmed that the animated-text functionality operated as expected.



IV.DISCUSSION

The results demonstrate the integration of multiple AI-based multimedia functions within a single web-based platform. The diffusion-model training and checkpoint integration provide the basis for text-to-image generation, while the image-enhancement, video-generation, animated-text, and voice-assistant modules extend the system's capabilities beyond static image creation.

The unified interface allows users to access different multimedia operations without switching between separate applications. The modular architecture also supports the independent development and maintenance of individual components. The twelve documented passing test cases provide evidence that the principal functionalities and their frontend-backend interactions operated successfully during the reported testing.

Despite these results, further evaluation is required to establish the quality and efficiency of the generated outputs. Quantitative image-quality measurements, repeated execution-time tests, independent evaluation of the intent classifier, and comparisons with relevant baseline methods would provide stronger evidence of system performance. The current implementation also faces limitations related to computational requirements, processing time, storage, and the scope of some multimedia functions. These limitations provide opportunities for future optimization and extension.

V. CONCLUSION

The project, “RANANO: Recreating Nano Banana Using Diffusion Model,” demonstrates the development of an integrated AI-based multimedia platform combining text-to-image generation, image enhancement, video generation, animated-text creation, and voice-assisted interaction within a common web interface.

The trained diffusion-model checkpoint was integrated into the application for image generation, and the major modules were evaluated through functional testing. All twelve documented test cases were recorded as passed, covering model loading, input validation, multimedia generation, output handling, and frontend–backend communication.

RANANO provides a unified environment for accessing several multimedia creation functions through a modular application architecture. Nevertheless, further work is needed to evaluate image quality, model generalization, computational efficiency, and performance under different operating conditions. Future development will focus on quantitative evaluation, processing-time optimization, broader testing, and improvements to the existing multimedia modules. Overall, the project establishes a foundation for further development of an integrated AI-assisted multimedia creation platform.

REFERENCES

1. J. Ho, A. Jain, and P. Abbeel, “Denoising Diffusion Probabilistic Models,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
2. Q. Nichol and P. Dhariwal, “Improved Denoising Diffusion Probabilistic Models,” *Proceedings of the 38th International Conference on Machine Learning (ICML)*, vol. 139, pp. 8162–8171, 2021.
3. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.
4. A. Radford et al., “Learning Transferable Visual Models from Natural Language Supervision,” *Proceedings of the 38th International Conference on Machine Learning (ICML)*, vol. 139, pp. 8748–8763, 2021.
5. C. Saharia et al., “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding,” arXiv preprint arXiv:2205.11487, 2022.
6. L. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
7. W. Zhang, Y. Zuo, and L. Zhang, “Deep Learning Based Image Super-Resolution and Enhancement: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
8. G. Bradski, “The OpenCV Library,” *Dr. Dobbs’s Journal of Software Tools*, vol. 25, no. 11, 2000.