NANYANG TECHNOLOGICAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Diffusion Models for Intelligent Image Editing and Inpainting

Lee Yu Quan

Supervisor: Prof Zhang Hanwang

School of Computer Science and Engineering

A Final Year Project report
presented to Nanyang Technological University
in partial fulfilment of the requirements for the
degree of Bachelor of Engineering

2025/2026

Acknowledgements

Over the course of two semesters working on this final year project, I would like to express my appreciation to everyone who has encouraged me and offered their guidance, helping make this project possible.

I would like to extend my sincere gratitude to my supervisor, Prof Zhang Hanwang, for granting me the freedom to steer the direction of this project. His trust and openness allowed me to explore a wide variety of diffusion models and techniques in the field of image generation, which greatly enriched both the project and my learning experience.

Lastly, I would like to thank my examiner, Prof (placeholder), for taking the time to review and evaluate this final year project.

Lee Yu Quan
March 2026

Contents

List of Figures

List of Tables

1 Introduction

1.1 Abstract

1.2 Background and Motivation

Large language models are the most discussed aspect of generative AI today, but image generation is not far behind. According to Grand View Research, the global AI image generator market was valued at USD 349.6 million in 2023 and is projected to grow at a compound annual growth rate (CAGR) of 17.7% to reach USD 1.08 billion by 2030 (Grand View Research, 2023). Yet, there remains a lack of comprehensive software that caters to this growing demand in an accessible manner.

Traditional methods in image inpainting, such as patch-based and exemplar-based approaches, have notable limitations in generating semantically meaningful content, particularly in high-resolution or complex scenarios (Ma et al., 2023). These methods often struggle with boundary artefacts when dealing with large masked regions due to insufficient constraints, resulting in visible seams and structurally inconsistent outputs (Ma et al., 2023). Such limitations create significant accessibility barriers, as current solutions frequently require extensive technical expertise and expensive software licences, effectively restricting advanced image editing capabilities to professional users.

The introduction of deep learning techniques, particularly diffusion models (Ho et al., 2020; Song et al., 2021; Rombach et al., 2022), has led to significant improvements in image generation quality and semantic comprehension, enabling capabilities that were previously difficult or impossible to automate. A detailed review of these developments is presented in Section 3.

Despite these breakthroughs, critical gaps persist between state-of-the-art image editing models and users' practical needs. Existing solutions face four key limitations: (1) fragmented ecosystems requiring users to switch between different applications for different editing tasks, (2) high complexity barriers that make advanced editing tools inaccessible to non-expert users, (3) reliance on command-line tools, Python programming, and GPU-equipped hardware, and (4) a lack of unified platforms that integrate multiple diffusion model capabilities within a single interface.

This project, DiffusionDesk, addresses these gaps by deploying a web-based platform that integrates diffusion models for inpainting, style transfer, and image restoration within a single, user-friendly interface. Developed as a Final Year Project at Nanyang Technological University under the supervision of Prof Zhang Hanwang, the application provides three core image editing capabilities:

1. **Inpainting** – removing or replacing objects within selected regions of an image using models such as Stable Diffusion, Stable Diffusion XL, Kandinsky, and FLUX.1 Fill.

2. **Style Transfer** – applying artistic styles such as anime, oil painting, and watercolour to images using diffusion-based image-to-image translation.

3. **Restoration** – enhancing image quality through face restoration (CodeFormer, GFP-GAN) and image upscaling (Real-ESRGAN).

By exposing these models through a FastAPI backend and a React-based browser frontend, DiffusionDesk aims to make diffusion-based image editing accessible to users without requiring direct interaction with the underlying models or command-line tools.

1.3 Project Objective

The objective of this project is to design and deploy a web-based image editing platform that leverages open-source diffusion models to provide intelligent inpainting, style transfer, and image restoration capabilities. The specific objectives are as follows:

1. To develop a responsive web application that integrates multiple diffusion models for image editing within a unified interface.

2. To implement an inpainting feature that enables users to selectively remove or replace objects in images using state-of-the-art diffusion models, including Stable Diffusion, Stable Diffusion XL, Kandinsky, and FLUX.1 Fill.

3. To implement a style transfer feature that allows users to apply artistic styles to images through diffusion-based image-to-image translation.

4. To implement an image restoration feature that enhances image quality through face restoration and upscaling using CodeFormer, GFPGAN, and Real-ESRGAN.

5. To design an intuitive user interface that enables non-expert users to perform advanced image editing tasks without requiring technical expertise in machine learning or programming.

6. To evaluate the system's performance through processing speed benchmarks, output quality assessments, and usability considerations.

1.4 Limitations

This project is subject to the following limitations:

1. **Open-source models only** – The application exclusively utilises open-source diffusion models available through the Hugging Face ecosystem. Proprietary or commercially licensed models are not included, which may limit the range of available capabilities compared to commercial solutions.

2. **GPU resource constraints** – Diffusion model inference is computationally intensive and requires GPU acceleration. The available GPU memory (VRAM) constrains the size and complexity of models that can be loaded simultaneously. Quantisation techniques (4-bit, 8-bit) are employed to mitigate this, but may result in slight quality degradation.

3. **Supported image formats** – The application supports JPEG, JPG, and PNG image formats only. Other formats such as TIFF, BMP, WebP, or RAW are not supported.

4. **No mobile application** – The platform is designed as a web application accessible through desktop and mobile browsers. A dedicated native mobile application is not within the project scope.

5. **No video processing** – The system processes individual images only. Video frame processing, video inpainting, or video style transfer are not supported.

6. **No 3D image manipulation** – The application is limited to 2D image editing. 3D reconstruction, 3D-aware editing, or depth-based manipulation are not included.

7. **Inference only** – The project focuses on model inference using pre-trained models. Model training, fine-tuning, or custom model development are outside the project scope.

1.5 Project Scope

The scope of this project encompasses the following:

1.5.1 In Scope

- Development of a web-based frontend using React, TypeScript, and Tailwind CSS that provides an intuitive user interface for all three editing features.
- Development of a backend API using FastAPI and PyTorch that serves diffusion model inference for inpainting, style transfer, and image restoration.
- Implementation of inpainting using Stable Diffusion Inpainting, Stable Diffusion XL Inpainting, Kandinsky Inpainting, and FLUX.1 Fill models.
- Implementation of style transfer using SDXL image-to-image generation with artistic style prompts.
- Implementation of image restoration using CodeFormer, GFPGAN (face restoration), and Real-ESRGAN (image upscaling).
- Support for JPEG, JPG, and PNG image formats.

- VRAM optimisation through model quantisation (4-bit, 8-bit) and CPU offloading to accommodate varying GPU configurations.
- Deployment and testing on cloud GPU environments such as Google Colab.

1.5.2 Out of Scope

- Native mobile application development.
- Video processing, video inpainting, or video style transfer.
- 3D image manipulation or depth-based editing.
- Model training, fine-tuning, or custom model development.
- User authentication, user account management, or multi-user collaboration features.
- Image formats other than JPEG, JPG, and PNG.

Success will be measured through processing speed benchmarks, output quality assessments, and user experience evaluation across the three core features.

2 Project Schedule

This section outlines the project timeline, work breakdown structure, and risk management plan for DiffusionDesk. The project spans two semesters of Academic Year 2025/2026, with key milestones aligned to the FYP submission deadlines.

2.1 Project Timeline

Table 1 presents the key milestones and deliverables for the project.

Table 1: Project Timeline and Milestones

Date	Week	Milestone
11 Aug 2025	Sem 1, Wk 1	FYP officially commences
1 Sep 2025	Sem 1, Wk 4	Submission of Project Plan to Supervisor
Oct 2025	Sem 1, Wk 8–10	Backend API and diffusion service implementation
Nov 2025	Sem 1, Wk 12–13	Frontend development and API integration
26 Jan 2026	Sem 2, Wk 3	Submission of Interim Report
Feb 2026	Sem 2, Wk 5–7	Feature refinement and testing on cloud GPU
23 Mar 2026	Sem 2, Wk 10	Submission of Final Report
17 Apr 2026	Sem 2, Wk 13	Submission of Amended Final Report
8–13 May 2026	–	Oral Presentation (20 min + 10 min Q&A)

2.2 Work Breakdown

This section provides a structured breakdown of the project activities, their descriptions, estimated effort, and dependencies. The work breakdown structure facilitates progress tracking and resource allocation throughout the development lifecycle.

Table 2: Work Breakdown Structure

ID	Activity	Description	Effort (Days)	Dependencies
1.1	Literature Review	Review foundational papers on diffusion models (DDPM, DDIM, LDM) and existing inpainting, style transfer, and restoration techniques.	10	NIL
1.2	Technology Evaluation	Evaluate and select appropriate frameworks, libraries, and pre-trained models from the Hugging Face ecosystem.	5	1.1

ID	Activity	Description	Effort (Days)	Dependencies
1.3	Requirements Specification	Define functional and non-functional requirements based on project objectives and supervisor feedback.	4	1.2
2.1	Backend Architecture Design	Design the FastAPI backend structure, including API endpoints, service layers, and model management strategy.	6	1.3
2.2	Inpainting Service Implementation	Implement the inpainting service supporting Stable Diffusion, SDXL, Kandinsky, and FLUX.1 Fill models with quantisation support.	15	2.1
2.3	Style Transfer Service Implementation	Implement the style transfer service using SDXL image-to-image generation with configurable style prompts and parameters.	10	2.1
2.4	Restoration Service Implementation	Implement face restoration (CodeFormer, GFPGAN) and image upscaling (Real-ESRGAN) services.	10	2.1
2.5	VRAM Optimisation	Implement model quantisation (4-bit, 8-bit) and CPU offloading strategies to support varying GPU configurations.	8	2.2, 2.3
3.1	Frontend UI Design	Design the user interface layout, including wireframes for the Home, Inpainting, Style Transfer, and Restoration tabs.	5	1.3
3.2	Frontend Implementation	Develop the React frontend with TypeScript and Tailwind CSS, implementing all UI components and tab navigation.	15	3.1

ID	Activity	Description	Effort (Days)	Dependencies
3.3	Canvas and Mask Drawing	Implement the interactive canvas component for users to draw inpainting masks on uploaded images.	8	3.2
3.4	API Integration	Connect the frontend to the backend API, implementing image upload, processing requests, and result display.	6	2.2, 2.3, 2.4, 3.2
4.1	Cloud Deployment Testing	Deploy and test the backend on Google Colab with ngrok tunnelling to validate GPU inference performance.	5	3.4
4.2	Feature Testing and Refinement	Conduct end-to-end testing of all features, addressing bugs and refining based on test results.	10	4.1
4.3	Performance Benchmarking	Measure and document inference times, memory usage, and output quality across different models and configurations.	5	4.2
5.1	Supervisor Meetings	Regular meetings with the supervisor to report progress, seek guidance, and align on project direction.	8	All Tasks
5.2	Interim Report Writing	Prepare and submit the interim report documenting progress, challenges, and preliminary results.	5	2.5, 3.4
5.3	Final Report Writing	Prepare the final report with comprehensive documentation of the system design, implementation, and evaluation.	15	4.3, 5.2
5.4	Oral Presentation Preparation	Prepare presentation slides and rehearse for the oral examination.	5	5.3

2.3 Risk Management

This section identifies potential risks that may impact the project's success and outlines mitigation strategies. Each risk is assessed based on its probability of occurrence, potential impact, and overall risk level. Proactive risk management ensures that challenges are anticipated and addressed promptly.

Table 3: Risk Management Plan

Risk Description	**Mitigation Strategy**	**Probability**	**Impact**	**Risk Level**
Insufficient GPU memory for large models	Implement model quantisation (4-bit, 8-bit) using bitsandbytes and enable CPU offloading. Prioritise smaller models (SD Inpainting) when VRAM is limited.	High	High	High
Model inference too slow for interactive use	Use DDIM sampling with reduced steps (20–50), enable attention slicing, and consider model caching to reduce repeated loading overhead.	Medium	High	Medium
Diffusion model produces low-quality or artefacted outputs	Fine-tune prompt engineering, adjust guidance scale and denoising strength parameters, and provide users with parameter controls for iterative refinement.	Medium	Medium	Medium
Breaking changes in Hugging Face Diffusers library	Pin specific library versions in requirements.txt and test compatibility before updating dependencies.	Medium	Medium	Medium

Risk Description	**Mitigation Strategy**	**Probability**	**Impact**	**Risk Level**
Cloud GPU availability constraints (e.g., Colab session limits)	Design the system for stateless operation, allowing sessions to be restarted without data loss. Document alternative deployment options (NTU HPC, local GPU).	Medium	Medium	Medium
Frontend-backend integration issues	Define clear API contracts using OpenAPI specifications, implement comprehensive error handling, and test integration incrementally throughout development.	Medium	High	Medium
Scope creep due to additional feature requests	Adhere strictly to the defined project scope. Evaluate new requests against timeline constraints and prioritise core features over enhancements.	Medium	Medium	Medium
Unfamiliarity with diffusion model architectures	Conduct thorough literature review early in the project. Leverage existing tutorials, documentation, and pretrained models to accelerate learning.	Low	High	Low

Risk Description	Mitigation Strategy	Probability	Impact	Risk Level
Personal time management challenges	Maintain a detailed project schedule with milestones. Allocate buffer time for unexpected delays and communicate proactively with the supervisor.	Low	High	Low

3 Literature Review

3.1 Diffusion Models

The foundation of modern image generation lies in diffusion models, which produce images through an iterative denoising process. This subsection reviews the key developments that underpin the models used in this project.

3.1.1 Denoising Diffusion Probabilistic Models (DDPM)

Ho et al. (2020) proposed Denoising Diffusion Probabilistic Models (DDPMs), which generate images by treating the process as a series of denoising steps grounded in nonequilibrium thermodynamics. The forward process gradually adds Gaussian noise to an image over T timesteps until the image becomes pure noise. The reverse process then learns to denoise step by step, recovering a clean image from random noise. DDPMs demonstrated image quality that surpassed the then-dominant Generative Adversarial Networks (GANs), producing diverse, high-fidelity samples without the training instability commonly associated with GANs. However, the original DDPM formulation required a large number of denoising steps (typically $T = 1000$), resulting in slow sampling speeds.

3.1.2 Denoising Diffusion Implicit Models (DDIM)

Song et al. (2021) addressed the slow sampling limitation of DDPMs by proposing Denoising Diffusion Implicit Models (DDIMs). DDIMs reformulate the reverse diffusion process as a non-Markovian process, meaning that each denoising step can depend on the original noisy input rather than solely on the immediately preceding step. This reformulation allows for deterministic sampling and, crucially, enables the use of a subsequence of only 20–100 steps while maintaining comparable image quality. The result is a 10–50$\times$ speedup over DDPMs, making diffusion-based generation significantly more practical for interactive applications.

3.1.3 Latent Diffusion Models (LDM)

Rombach et al. (2022) proposed Latent Diffusion Models (LDMs), which achieved an optimal balance between generative quality and computational efficiency. Rather than performing diffusion directly in pixel space, LDMs first encode images into a lower-dimensional latent representation using a pre-trained autoencoder, then apply the diffusion process within this compressed latent space. This approach alleviates critical computational bottlenecks, substantially reducing memory and computation requirements while preserving high image quality. LDMs form the basis of the widely adopted Stable Diffusion family of models, including the inpainting and image-to-image variants used in this project.

3.2 Inpainting Techniques

3.2.1 Stable Diffusion Inpainting

3.2.2 Stable Diffusion XL Inpainting

3.2.3 Kandinsky Inpainting

3.2.4 FLUX.1 Fill

3.3 Style Transfer

3.3.1 Traditional Neural Style Transfer

3.3.2 Diffusion-Based Style Transfer

3.3.3 Style Prompting Strategies

3.4 Image Restoration

3.4.1 Face Restoration

3.4.1.1 CodeFormer

3.4.1.2 GFPGAN

3.4.2 Image Upscaling

3.4.2.1 Real-ESRGAN

3.5 Technology Stack Considerations

This section reviews and justifies the technology choices for developing DiffusionDesk, considering factors such as performance, ecosystem support, developer experience, and suitability for machine learning workloads.

3.5.1 Frontend Framework Selection

3.5.2 Backend Framework Selection

3.5.3 Machine Learning Framework and Libraries

3.5.4 Deployment Considerations

4 Software Requirements

4.1 Use Case Diagram

4.1.1 Use Case Descriptions

4.2 Functional and Non-Functional Requirements

4.2.1 Functional Requirements

4.2.2 Non-Functional Requirements

5 Planning and Design

5.1 Project Development Methodology

5.2 System Architecture

5.3 User Interface Wireframe

6 Implementation

6.1 Backend Development

6.1.1 Project Structure

6.1.2 API Design

6.1.3 Inpainting Service

6.1.4 Style Transfer Service

6.1.5 Restoration Service

6.1.6 Model Management and VRAM Optimization

6.2 Frontend Development

6.2.1 Project Structure

6.2.2 User Interface Design

6.2.3 Canvas and Mask Drawing

6.2.4 API Integration

7 Project Difficulties and Learning Outcomes

7.1 Project Difficulties

7.2 Learning Outcomes

8 Future Implementation

Bibliography

Grand View Research (2023). AI image generator market size, share & trends analysis report, 2024–2030. Accessed: 2026-02-02.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ma, X., Zhou, X., Huang, H., Jia, G., Wang, Y., Chen, X., and Chen, C. (2023). Uncertainty-aware image inpainting with adaptive feedback network. *Expert Systems with Applications*, 235:121148.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Song, J., Meng, C., and Ermon, S. (2021). Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*.