Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

Text-Driven Image-to-Image Generation

July 2024

Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

Text-Driven Image-to-Image Generation

By:

Amir Moris Habib	[Computer Science]
Verina Gad Soliman	[Computer Science]
Mina Girgis Alfy	[Computer Science]
Carolina George Hana	[Computer Science]
Maria Tawfek Waheb	[Computer Science]

Under Supervision of:

Prof. Dr. Abdel-Badeeh Salem [BSc, MSc, PhD],
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University

Dr. Hanan Hindy [BSc, MSc, PhD],
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

A.L. Ghada Elnahas [BSc, MSc],
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

A.L. Yomna Ahmed [BSc, MSc],
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

Acknowledgment

First and foremost, we give all praise and thanks to ALLAH, whose grace and blessings have enabled us to complete this work. We hope that this effort is accepted and blessed.

We express our deepest gratitude to our parents and families for their constant and unwavering support, love, and encouragement throughout our years of study. We hope to make them proud and repay their kindness.

We profoundly extend our heartfelt appreciation to our supervisors, Prof. Dr. Abdel-Badeeh Salem, Dr. Hanan Hindy, A.L. Ghada Elnahas, and A.L. Yomna Ahmed, for their insightful and invaluable guidance, patience, and expertise throughout our thesis journey.

Finally, we would like to thank our friends and everyone who has provided us with support and encouragement along the way. Your belief in us has been a source of inspiration.

Abstract

This project focuses on creating a model that updates images based on text input, which is essential for social media, advertising, and content creation. Users can upload an image and provide text instructions for changes, which can include adding annotations, modifying elements, or adding new visual information. The system processes these instructions to update the image while maintaining high visual quality.

Key features include an easy-to-use interface, advanced algorithms for understanding and applying text instructions, and changing visual elements as needed. Among experiments with state-of-the-art text-driven image-to-image models, a significant obstacle was encountered: the substantial memory and GPU resources required. However, two models stood out as efficient and lightweight.

The first model experimented with was the SINE model, which consists of two main parts: training the text-to-image diffusion model on the dataset and fine-tuning it on the input image. The output image is generated using a linear combination function of both parts. Despite being lighter-weight, the fine-tuned text-to-image diffusion model still required extensive resources to run.

Ultimately, the InstructPix2Pix model was utilized, proving to be efficient in both performance and memory usage. A notable challenge was the lack of domain-specific datasets in the text-driven image-to-image field. To address this, the focus was on generating a domain-specific dataset, choosing "products" as the domain of interest. The dataset generation process included two sub-processes: textual data generation and image data generation.

Similar works usually use a large language model (LLM) like GPT-3 to generate the textual dataset. However, due to the high costs associated with its use, a simpler approach was opted for a hardcoded text synthesizer. Although this method is considered naive, it provided high-quality text prompts that were successfully used to create the necessary dataset. Testing shows that the system works reliably, producing clear and accurate updates based on various image types and text instructions.

In conclusion, this project provides a simple and effective way to update images using text instructions and introduces a specialized products dataset, which is rare in the field. It is beneficial for content creators, educators, marketers, and social media users, helping them create customized and engaging visual content.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
CFG	Classifier Free Guidance
GANs	Generative Adversarial Networks
LLM	Large Language Model
NLP	Natural Language Processing
NNs	Neural Networks
P2P	Prompt-to-Prompt model
SINE	SINgle Image Editing with Text-to-Image Diffusion Models
UI	User Interface

1- Introduction

1.1 Motivation

The ability to seamlessly modify images based on textual descriptions represents a groundbreaking advancement in the domain of artificial intelligence. This project, which focuses on updating existing images according to specified textual changes, holds immense potential across various domains such as advertising, e-commerce, entertainment, social media, digital art, and graphic design.

Advertising: In the fast-paced world of advertising, the need for adaptable and dynamic visual content is ever-growing. This technology allows advertisers to update campaign visuals quickly and efficiently by simply altering the text. For instance, a seasonal advertisement can be easily modified to reflect a new sale or product feature without needing to recreate the entire image from scratch, thereby saving time and resources.

E-Commerce: In e-commerce, product images are a critical component of the customer experience. With this technology, merchants can update product visuals to reflect new descriptions, variations, or promotional offers. This ensures that product listings remain current and visually appealing, enhancing the shopping experience and potentially increasing sales.

Entertainment and Social Media: The entertainment industry and social media platforms thrive on engaging and up-to-date content. This project enables content creators to adapt existing visuals to new contexts or narratives by changing the accompanying text. For example, movie posters or social media graphics can be quickly updated to include new release dates, event details, or promotional messages, maintaining relevance and interest.

Digital Art and Graphic Design: For digital artists and graphic designers, the ability to modify images based on textual input provides a powerful tool for creative expression. It allows artists to iterate on their work easily, incorporating feedback and new ideas without starting from scratch. This flexibility fosters greater creativity and efficiency in the design process

In conclusion, the motivation behind this Text-Driven Image Modification project lies in its potential to revolutionize how images are updated and adapted across various industries. By leveraging the power of text to drive visual changes, we can create a more dynamic, efficient, and accessible approach to image modification, ultimately enhancing the way visual content is created and experienced.

1.2 Problem Definition

Our project focuses on generating new images based on a given guidance image and a text prompt. Most Image-To-Image applications in the market do not focus on a specified domain which leads to very low accuracies and the generation of images with bad quality. Additionally, there is no readily available dataset for products in the Image-To-Image translation domain. By concentrating on a specific domain which is the product domain, we aim to enhance the accuracy and quality of the generated images. This focus would be particularly useful for applications in e-commerce, marketing, and inventory management.

1.3 Objective

In this project, we had the following main objectives:

1. Automate Image Generation:

Develop an advanced system that automates the process of generating and editing product images based on user-provided text prompts, thereby reducing manual effort and increasing efficiency in content creation.

2. Domain-Specific Model Enhancement:

Focus on fine-tuning the Text-driven image-to-image translation model specifically for product images used in marketing for brands. This specialization aims to improve the model's accuracy and effectiveness within this slot, ensuring high-quality and contextually relevant outputs.

3. Object-Focused Image Editing:

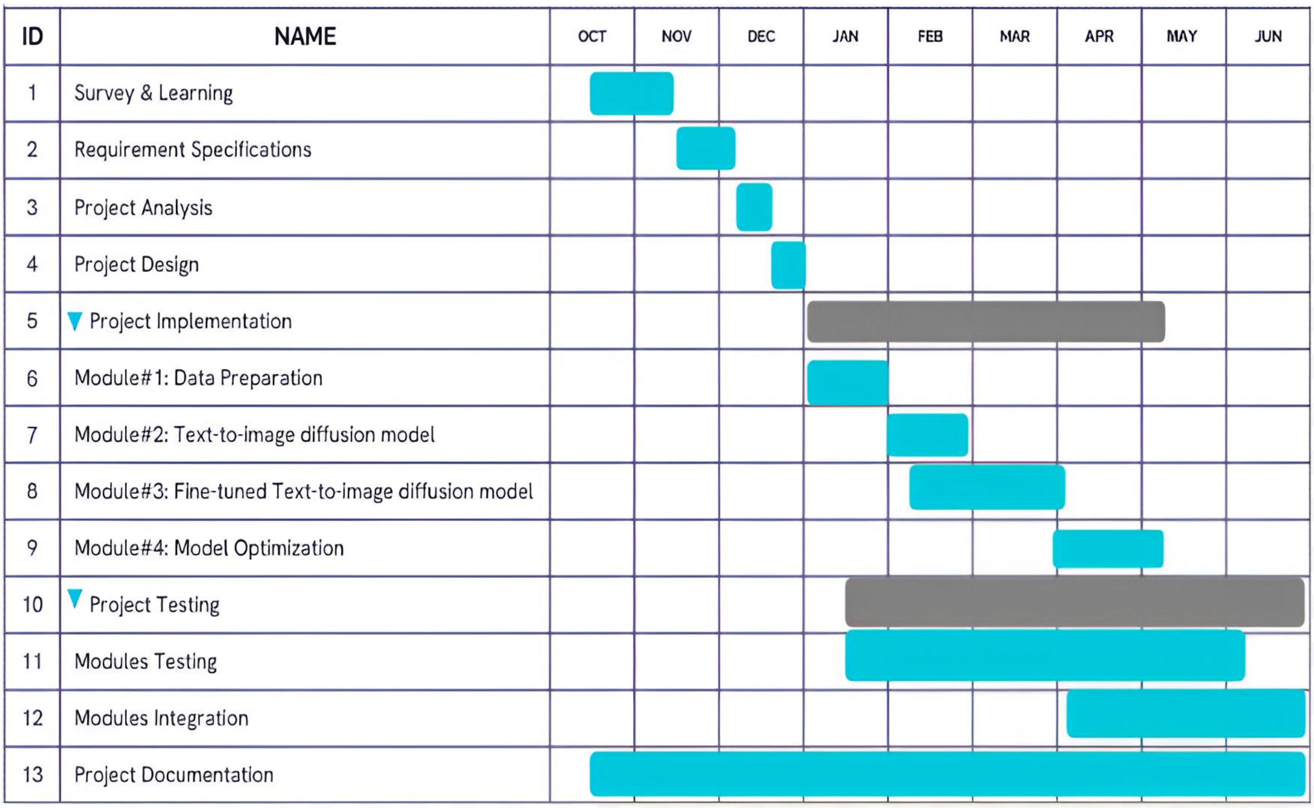
Implement techniques that preserve the features of the main object in an image while creating entirely different perspectives, backgrounds, or textures around it. This approach allows for significant customization and variation in image presentation, enhancing the visual appeal for marketing purposes.

4. Specialized Dataset Generation:

Generate paired texts and images to utilize for fine-tuning on a specific domain. Due to the lack of existing datasets focused on product images, we created our own custom dataset tailored to the product domain. This ensures the model is trained with relevant and high-quality data, enhancing its performance for specific industry needs.

1.4 Time Plan

Figure 1.1 illustrates the project's overall timeline, depicting key stages and their durations from the initial literature review to the final product development. Note the overlapping tasks, with project documentation being continuously updated throughout.



ID	NAME	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN
1	Survey & Learning	■	■							
2	Requirement Specifications		■	■						
3	Project Analysis			■						
4	Project Design			■						
5	▼ Project Implementation				■	■	■	■	■	
6	Module#1: Data Preparation				■					
7	Module#2: Text-to-image diffusion model					■				
8	Module#3: Fine-tuned Text-to-image diffusion model					■	■	■		
9	Module#4: Model Optimization						■	■	■	
10	▼ Project Testing				■	■	■	■	■	■
11	Modules Testing				■	■	■	■	■	■
12	Modules Integration							■	■	■
13	Project Documentation	■	■	■	■	■	■	■	■	■

Figure 1.1 Time Plan

1.5 Document Organization

- **Chapter 2:** This chapter discusses the project's background, including its scientific basis and intended usage field. It provides a theoretical background and a brief description of other related work.

- **Chapter 3:** This chapter discusses the system architecture, including a description of each of its components and the 3-tier architecture used. It also covers the architecture of the deep learning models employed. Additionally, the chapter describes the system’s intended users and their characteristics, including the basic knowledge required for users to benefit from the project.

- **Chapter 4:** This chapter explains the implementation and testing process of the project. It provides detailed implementation information, highlights the experiments conducted, and outlines the testing procedures followed.

- **Chapter 5:** This chapter provides a detailed user manual outlining how to use the system, complete with a step-by-step screenshot guide.

- **Chapter 6:** This chapter concludes the documentation and proposes future improvements aimed at maximizing the project’s potential.

2- Background

2.1 Project Overview

The project focuses on advancing text-driven image-to-image generation technology, which automates image modification and creation based on textual descriptions. It harnesses cutting-edge deep learning models and sophisticated algorithms, with particular emphasis on diffusion models and stable diffusion techniques.

Technologically, the project builds upon deep learning architectures such as text-to-image generation models enhanced by GPT-3-based approaches. Diffusion models play a crucial role, utilizing iterative noise manipulation in latent space to refine images and ensure realistic outputs. Additionally, Conditional Generative Adversarial Networks (GANs) contribute to precise image domain mapping, enabling accurate transformations based on textual cues.

This technology finds diverse applications across various industries. In advertising, it enables dynamic modification of visual campaigns by adapting images in real-time to align with new marketing messages or seasonal themes. In e-commerce, the system enhances product listings by generating images that accurately reflect updated descriptions, features, or promotional offers. Moreover, entertainment and social media, it facilitates the creation of compelling content by seamlessly modifying existing visuals to fit new narratives or current events.

Scientifically, the project builds upon recent advancements in Artificial Intelligence (AI) and image processing techniques. It preserves image fidelity while integrating textual edits for coherence and quality. Natural Language Processing (NLP) algorithms play a critical role in accurately understanding and processing textual inputs, driving precise image modifications.

2.2 Theoretical Background

According to the latest advancements and research, image generation models can be categorized into four types based on differences in input, output, method, or architecture, these types are:

- Segmentation mask image-to-image generation [1]: It is a process that involves generating images from segmentation masks while incorporating textual descriptions. This task combines both text and segmentation masks as inputs to generate corresponding realistic images that align with the textual descriptions. An example is shown in Figure 2.1.

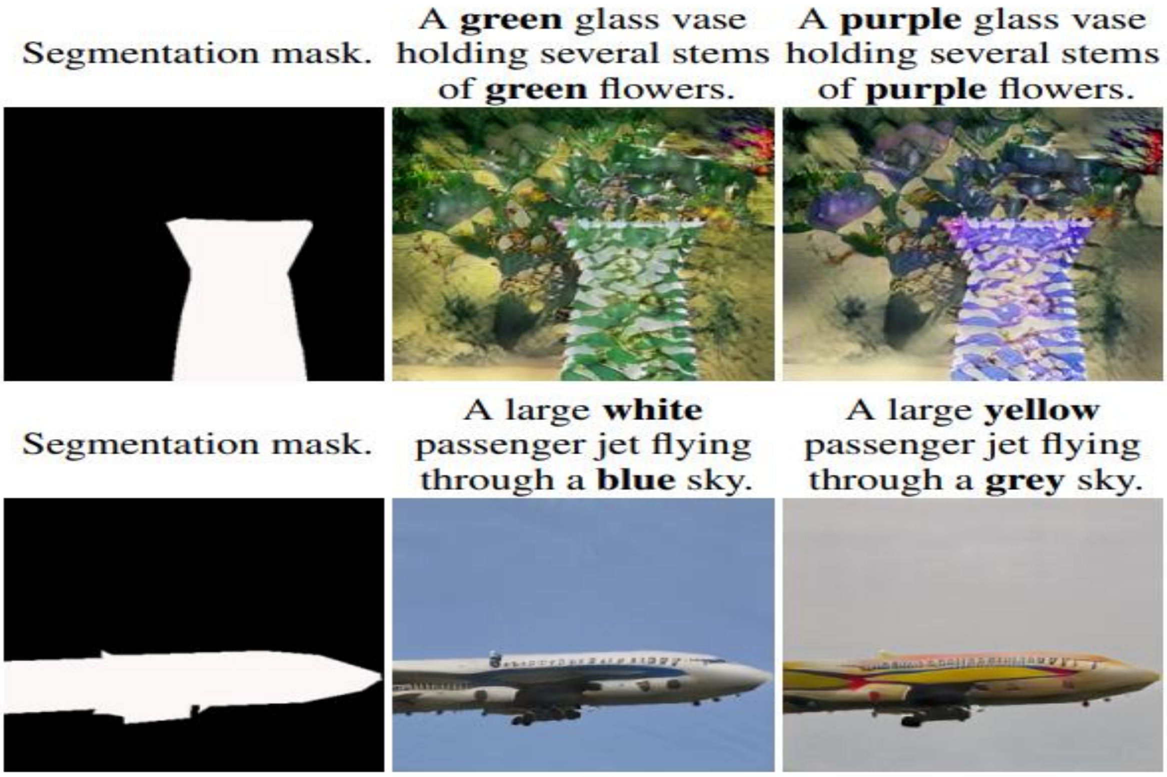

Segmentation mask.
A **green** glass vase holding several stems of **green** flowers.
A **purple** glass vase holding several stems of **purple** flowers.
Segmentation mask.
A large **white** passenger jet flying through a **blue** sky.
A large **yellow** passenger jet flying through a **grey** sky.

Figure 2.1: Segmentation mask image-to-image generation.

- Text-to-image Generation: Text-to-image generative models [2, 3, 4] are capable of producing high-quality images using user-provided text prompts. In recent times, diffusion-based models have shown strong performance in text-to-image tasks. Stable Diffusion [5, 6, 7] proposes conducting the diffusion process in latent space rather than pixel space, which reduces the sampling steps without compromising image quality. An example is shown in Figure 2.2.

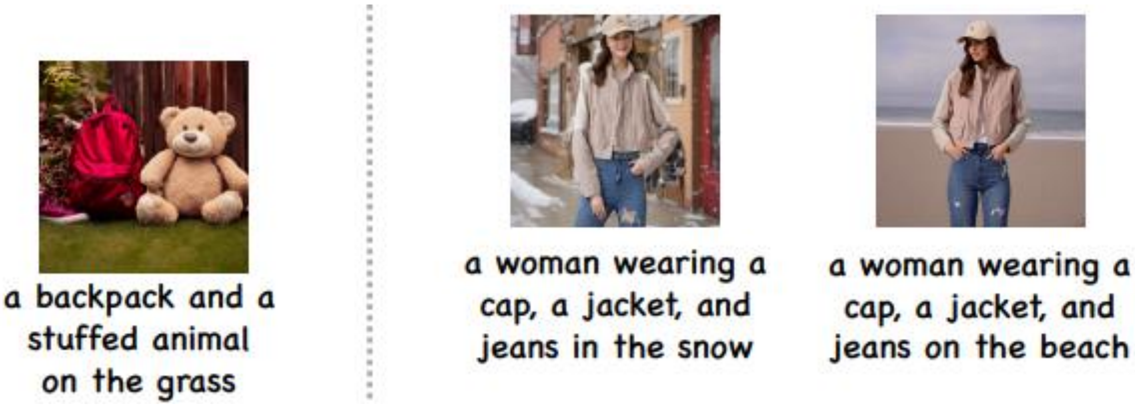

a backpack and a stuffed animal on the grass
a woman wearing a cap, a jacket, and jeans in the snow
a woman wearing a cap, a jacket, and jeans on the beach

Figure 2.2 Text-to-image Generation.

- Image-to-image generation: Image-to-image translation models [8, 9] have emerged as a prominent subfield within computer vision, enabling the manipulation and transformation of existing imagery. These models [10] leverage deep learning architectures to map an input image from a source domain to a target domain, resulting in a semantically equivalent image with desired alterations. This technology unlocks a plethora of applications. An example is shown in Figure 2.3.

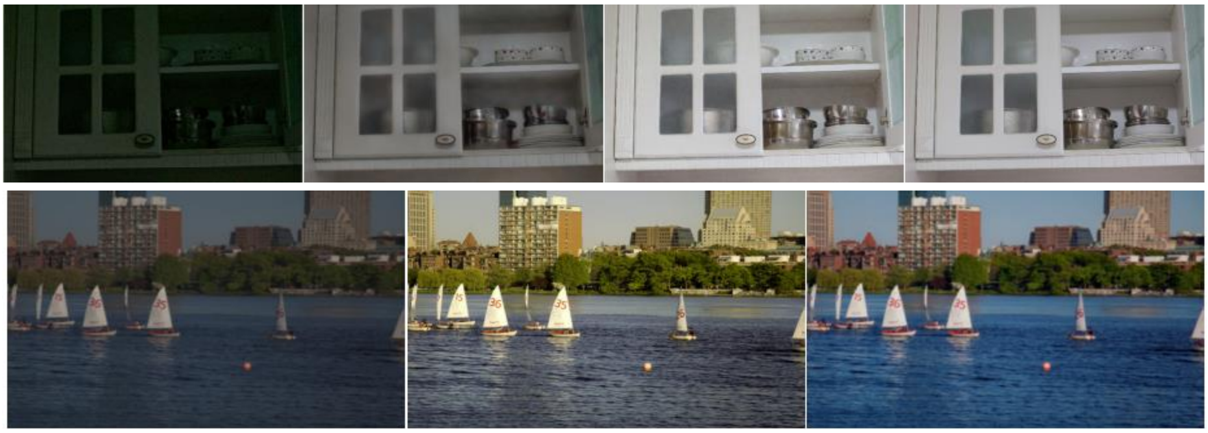


Figure 2.3: Image-to-Image Generation

- Text-driven image-to-image generation: Recently developed large text-to-image models have shown unprecedented capabilities, by enabling high-quality and diverse synthesis of images based on a text prompt written in natural language. Some models [11] enable such tasks by taking more than one input image and an edit instruction to generate an output image for different tasks like art rendition, view synthesis, and property modification As shown in Figure 2.4. Other models [13, 14, 16] enable generating output image by using only one input image provided by the user in addition to the edit instruction, also there are subject-driven models [11, 12, 15] that take one input image and subject text to generate the output image. An example is shown in Figure 2.5.

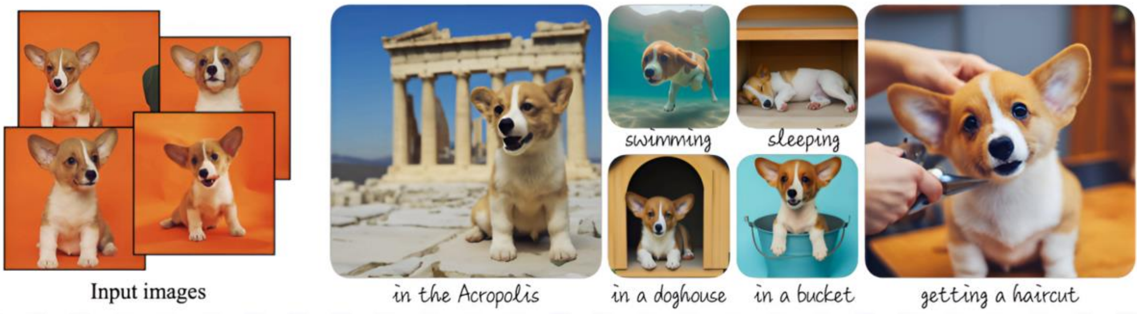


Figure 2.4: Text-driven image-to-image generation with multiple input images.

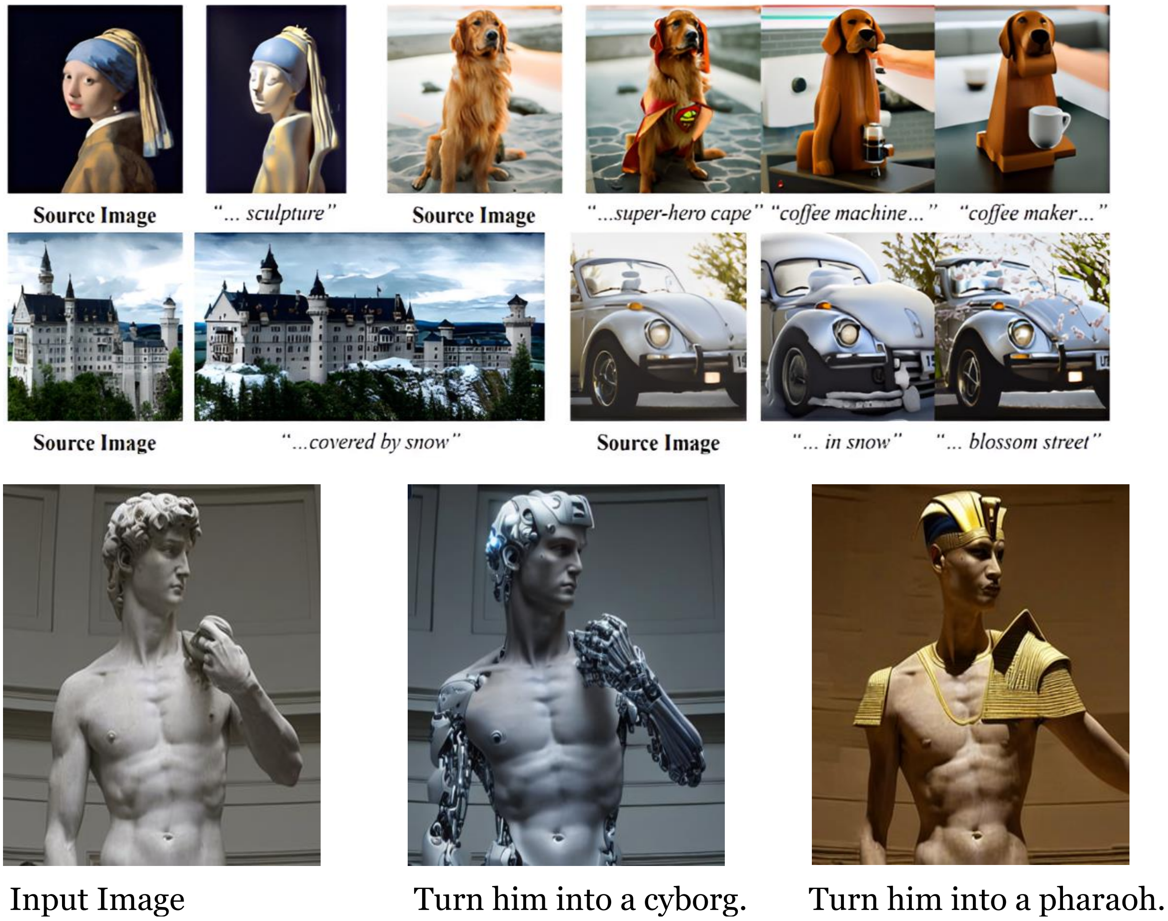


Figure 2.5: Text-driven Image-to-Image generation with one input image.

Diffusion models [18] have emerged as powerful tools in the realm of generative models, particularly for tasks involving image synthesis and translation. These models operate by iteratively denoising a variable, usually starting from random noise, until a coherent image is formed. Text-to-image models, a notable application of diffusion models, leverage this iterative refinement process to generate images based on textual descriptions. Within the context of text-driven image-to-image translation, diffusion models play a crucial role. They enable the transformation of an input image into a modified version that aligns with a given textual prompt. This approach involves conditioning the diffusion process not only on the original image but also on the semantic content provided by the text. By guiding the model through the denoising steps, it ensures that the output image retains the structural elements of the original while incorporating the attributes described by the text, thus achieving a seamless and coherent translation that reflects the specified modifications.

The main field of our project is Text-driven image-to-image generation, As we utilize the edit instruction and the input image entered by the user to generate the output image, the evolution in image-to-image generation models has progressed from the use of traditional Neural Networks (NNs) to (GANs) and more recently to diffusion models, each bringing distinct approaches and advancements to the field. NNs laid the groundwork with basic image processing capabilities, (GANs) and introduced a framework where two networks, a generator, and a discriminator, compete against each other to improve image quality, and diffusion models have advanced the state-of-the-art by iteratively refining images through a process of adding and then removing noise, leading to highly realistic and detailed outputs.

2.3 Related Work

In this section, we will focus on reviewing significant papers that are related to our project's field of study.

Ruiz et al. [11] proposed "DreamBooth" which addresses a key limitation in text-to-image generation: Traditional models often struggle to accurately render specific subjects or objects that are not well-represented in their training data. Alternatively, DreamBooth provides the ability to customize and fine-tune models to generate images of particular subjects based on minimal data through a fine-tuning technique for text-to-image diffusion models that allows for precise and personalized generation of images.

The method fine-tunes pre-trained diffusion models on a small set of images of a given subject. By doing so, it enables the model to generate new images of the subject in various contexts and styles, guided by textual descriptions. This approach leverages the rich representations learned by large-scale diffusion models while introducing subject-specific adjustments that enhance the fidelity and specificity of the generated images.

Despite its advancements, DreamBooth has certain limitations. One primary limitation is the dependency on a small set of images for fine-tuning. While this reduces the need for large datasets, it may still pose challenges when the available images are not representative enough of the subject's variability. Additionally, the fine-tuning process can be computationally expensive and time-consuming, potentially limiting its applicability in real-time or resource-constrained environments.

Furthermore, the method may face difficulties in generalizing to subjects or styles that are significantly different from those seen during pre-training, which can impact the quality and relevance of the generated images in such cases.

Zhang et al. [14] proposed "SINE" which addresses the challenge of editing specific attributes of a single image using textual descriptions, leveraging the capabilities of diffusion models to achieve high-fidelity edits.

This approach is particularly useful for applications where only a single image of the subject is available, and exact text-guided modifications are required. Building upon the foundational capabilities of diffusion models, SINE employs techniques such as Classifier-Free Guidance (CFG) and latent code optimization to iteratively refine the image based on the given textual instructions.

The key innovation in SINE lies in its ability to focus edits on specific regions of the image while preserving the overall structure and context. This is facilitated through diffusion model methodologies that utilize attention mechanisms to focus on specific parts of an image during the editing process, and region-specific loss functions to ensure that changes made to targeted areas align with the intended modifications described in textual input. The model utilizes hybrid feature blending such as Model-based CFG and patch-based regularization to generate high-quality edits that accurately reflect the textual descriptions, making it a powerful tool for tasks such as attribute modification, style transfer, and content alteration.

Despite its strengths, SINE faces limitations such as the need for fine-tuning, which demands substantial computational resources and time. The method's effectiveness is sensitive to the choice of guidance steps and weights, and there is a trade-off between edit fidelity and generalization capability. The process can sometimes introduce artifacts, especially when balancing high fidelity and strong edits, and the quality of the edited images depends significantly on the initial input image and fine-tuning data.

Hertz et al [17]. proposed "Prompt-to-Prompt (P2P)" which focuses on editing images by simply modifying the textual prompt using a technique called word-swap. The idea is to leverage the "cross-attention" mechanism in text-to-image models, which connects the textual prompt to different parts of the image during generation. By fine-tuning the prompt, P2P aims to control the edits without needing masks. It takes the input image, input text that describes the input image, and output text that describes the desired output image. Using the word-swap technique, it learns which attention map in the input image to change or manipulate.

Brooks et al [16]. proposed "InstructPix2Pix" that leverages a conditional diffusion model rather than (GANs). This model utilizes an input image and edit text instruction to produce an edited image, without requiring a separate discriminator for authenticity evaluation.

The distinctiveness of the Pix2Pix approach lies in its conditioning mechanism, where both the generator and discriminator are conditioned on the input image, ensuring that the generated output is not only realistic but also contextually aligned with the input. This differentiates Pix2Pix from traditional (GANs), which typically generate images from random noise. The model is trained on an AI-generated paired dataset, comprising input images and their corresponding target images, allowing it to learn a direct mapping from input to output. Such an approach is particularly effective in tasks such as transforming sketches into photographs, colorizing grayscale images, and converting aerial photographs into maps, where the structured relationship between the input and output pairs is leveraged to produce high-fidelity translations.

Pix2Pix introduces a new way of dataset creation as they used gpt3 to generate a textual dataset to obtain input image description, edit instruction that is performed on the input image description, output image description, then they use stable diffusion model to generate the input image and they by using input image, input image description they use P2P to generate the output image.

Pix2Pix, despite its strengths, exhibits limitations. AI-generated training data may introduce biases and limit the model's ability to handle unseen concepts. Additionally, natural language ambiguity can lead to misinterpretations of editing instructions. Furthermore, the model might struggle with highly specific edits or complex scene manipulations due to limitations in semantic understanding. Future work should explore mitigating these issues through bias detection and improved scene reasoning capabilities.

Table 2.1 shows a comparison between the discussed papers and provides a summary of the techniques and datasets used.

Paper	Technique	Dataset
DreamBooth [11]	Generative Diffusion models (Imagen - Stable diffusion model) 1. Fine-tuning 2. class-specific prior-preservation loss	MS COCO LAION-Aesthetics V2 6.5+
SINE [14]	The linear combination function operates between a text-to-image diffusion model trained on a dataset and a diffusion model trained on user-entered inputs.	LAION dataset flickr.com unsplash.com
Prompt-to-Prompt [17]	Image editing with cross-attention control and word swap technique.	COCO Dataset PIE-Bench
InstructPix2Pix [16]	GANs & fine-tuning a text-to-image diffusion model.	Ai-Generated

Table 2.1: Related Work Comparison

3- System Architecture and Methods

3.1 System Architecture

The system architecture diagram (Figure 3.1) showcases that our framework is divided into three layers: the Presentation Layer, Logic Layer, and Data Layer. The Presentation Layer includes the application interface where users upload input images and provide edit instructions, with the processed output image being displayed here as well. The Logic Layer handles the core processing, beginning with data generation, followed by the text-to-image model which uses the generated data and user inputs to create the desired images, and then applies CFG to balance text and image effects. The Data Layer stores the dataset, including input images, output images, and edit instructions, generated using AI techniques, which the Logic Layer uses for model training and image processing. The flow of data between these layers ensures a seamless transformation of user inputs into the final edited images.

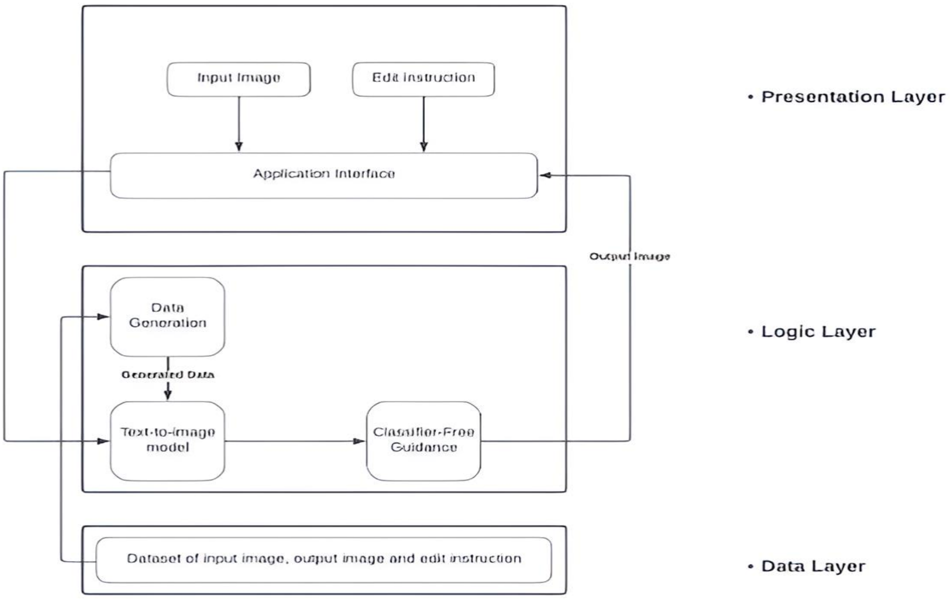


Figure 3.1: System Architecture

- **Presentation Layer:**

  The presentation layer, or user interface (UI) layer, serves as the front end of the application where users interact with the system. It is responsible for receiving inputs from users and displaying outputs. In the context of image processing, the presentation layer captures an image input and accompanying prompt text from the user. These inputs are then passed to the processing layer for modification. After processing, the resulting output image, reflecting the desired modifications specified by the user, is displayed back to the user through the presentation layer.

- **Logic Layer:**

  The Logic Layer consists of three main parts, each playing a crucial role in processing and transforming images based on user inputs.

  - **Data Generation**: This part uses AI to create a dataset. Each row in the dataset includes an input image, an edited caption, and an input caption. This generated data provides the foundational training material for the text-to-image model, ensuring it has diverse and relevant examples to learn from.

  - **Text-to-Image Model**: The text-to-image model utilizes the stable diffusion model shown in Figure 3.2, which is trained using the data from the previous step. Once trained, the model can take the input image and prompt text from the Presentation Layer and generate a new image with the desired modifications. This part of the Logic Layer is crucial for interpreting user instructions and transforming them into visual content.

  - **Classifier-Free Guidance:** The third part is the classifier-free guidance, which balances the influence of the text instructions and image features. This mechanism ensures that the generated image accurately reflects the user’s specifications, maintaining a harmonious integration of the text and image effects. This balance is essential for producing coherent and visually appealing results.

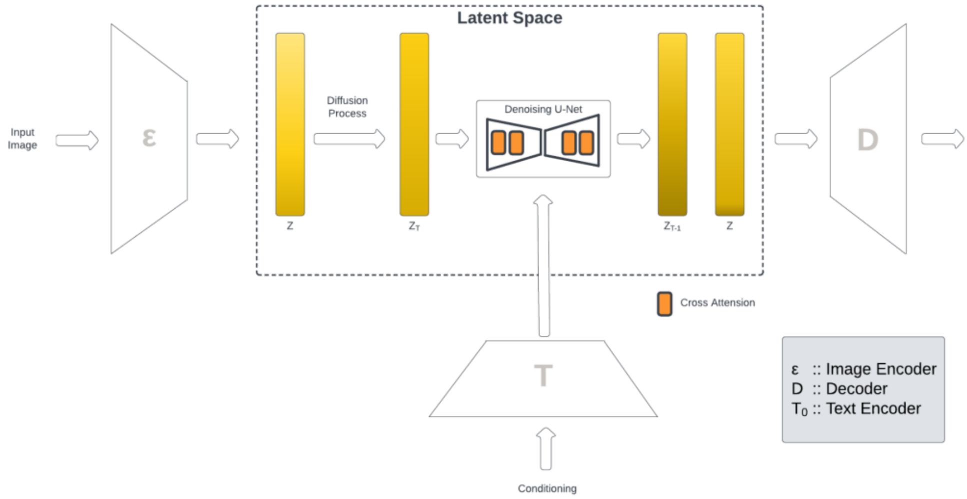


Figure 3.2: Diffusion Model

The diffusion model in latent space with conditioning involves encoding the input image and text instructions, introducing noise through a diffusion process, and then iteratively denoising the latent representation with guidance from the text instructions. The text encoder processes the edit instructions and provides the necessary conditioning information. The denoising process is carried out by a U-Net, which efficiently removes noise and refines the latent representation. The final clean latent representation is decoded back into an image that reflects the user-specified edits. The cross-attention mechanism plays a crucial role in integrating textual information during the denoising process, ensuring the edits are applied effectively and harmoniously.

- **Data Layer:**
  The data layer is responsible for managing the custom-generated dataset, which includes input images, output images, and edit instructions for each row. This dataset was created using AI due to the lack of suitable existing datasets. The data layer handles the storage, retrieval, and maintenance of this dataset, ensuring it is accessible and up-to-date for use by the logic layer. The process of generating this dataset will be discussed in subsequent sections.

3.2 System Users

A. Intended Users:

Our application is tailored to meet the needs of marketing professionals looking to create compelling visuals effortlessly. Whether developing social media campaigns, crafting ads, or designing promotional materials, our platform simplifies the process of transforming text and image prompts into visually captivating content.

B. User Characteristics

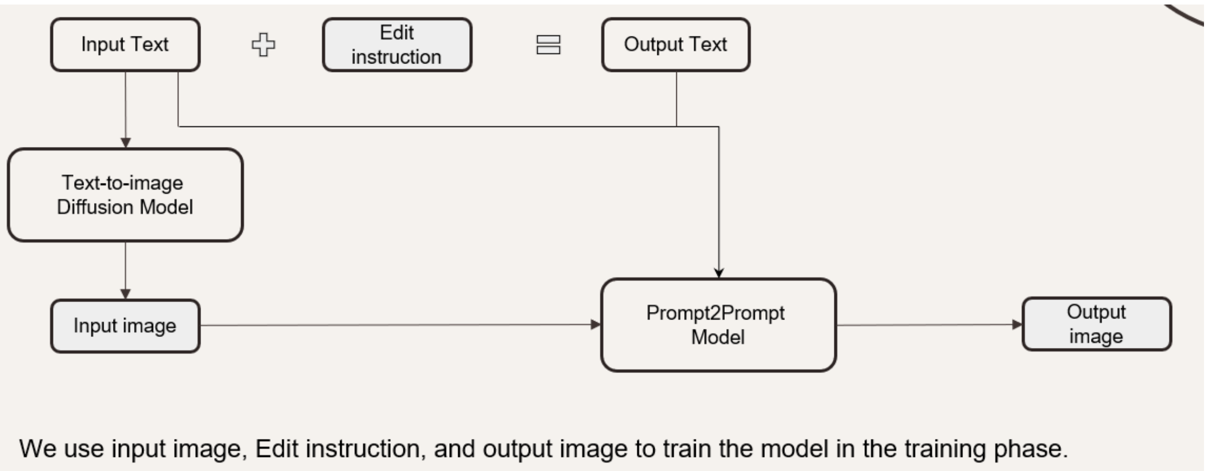
This project is designed to be user-friendly, requiring no specific skills. Anyone with an image and text prompt can easily utilize the tool to generate new images based on their inputs. Users can use our application with default parameters for simplicity. Additionally, there is an option to change the parameters for those with experience, allowing for more customized and advanced image generation.

4- Implementation and Testing

4.1 Dataset Generation

We noticed a lack of availability for domain-specific datasets in the Text-driven Image-To-Image field. We addressed this issue by focusing on the generation of a domain-specified dataset, we chose “products” to be our domain of interest.

The process of dataset generation consists of two sub-processes. Textual Data Generation and Image Data Generation. Figure 4.1 shows the complete dataset generation pipeline.



We use input image, Edit instruction, and output image to train the model in the training phase.

Figure 4.1: Dataset Generation

A. *Textual Data Generation:* We use a hardcoded text synthesizer to scrape the website Unsplash.com to obtain image descriptions of various products, which can be categorized into fashion, electronics, furniture, etc. We then apply different editing instructions to these descriptions, allowing for approximately 20 different editing effects on the input image descriptions, so we end up obtaining input image description, edit instruction, and output image description.

B. *Image Data Generation:* We use the input image description to generate the input image itself by utilizing a text-to-image diffusion model. Then we use P2P [17] to generate the output image, so for each row in the textual Data Generation process, we can obtain multiple rows in the Image Data generation as the diffusion model & P2P enables us to generate different pairs of input image & output image.

By combining process A with process B, we end up obtaining 5 columns which are:

- Input image description: used to generate input image.
- Edit instruction: used to generate output image description and used in the training process.
- Output image description: used to generate the output image.
- Input image: used to generate output image and used in the training process.
- Output image: used in the training process as the ground truth of the Image-To-Image model.

So, we only need the input image, edit instruction, and output image to fine-tune the model in the training process. Figure 4.2 illustrates various input images along with their text descriptions, the applied edit instructions, and the resulting output images.

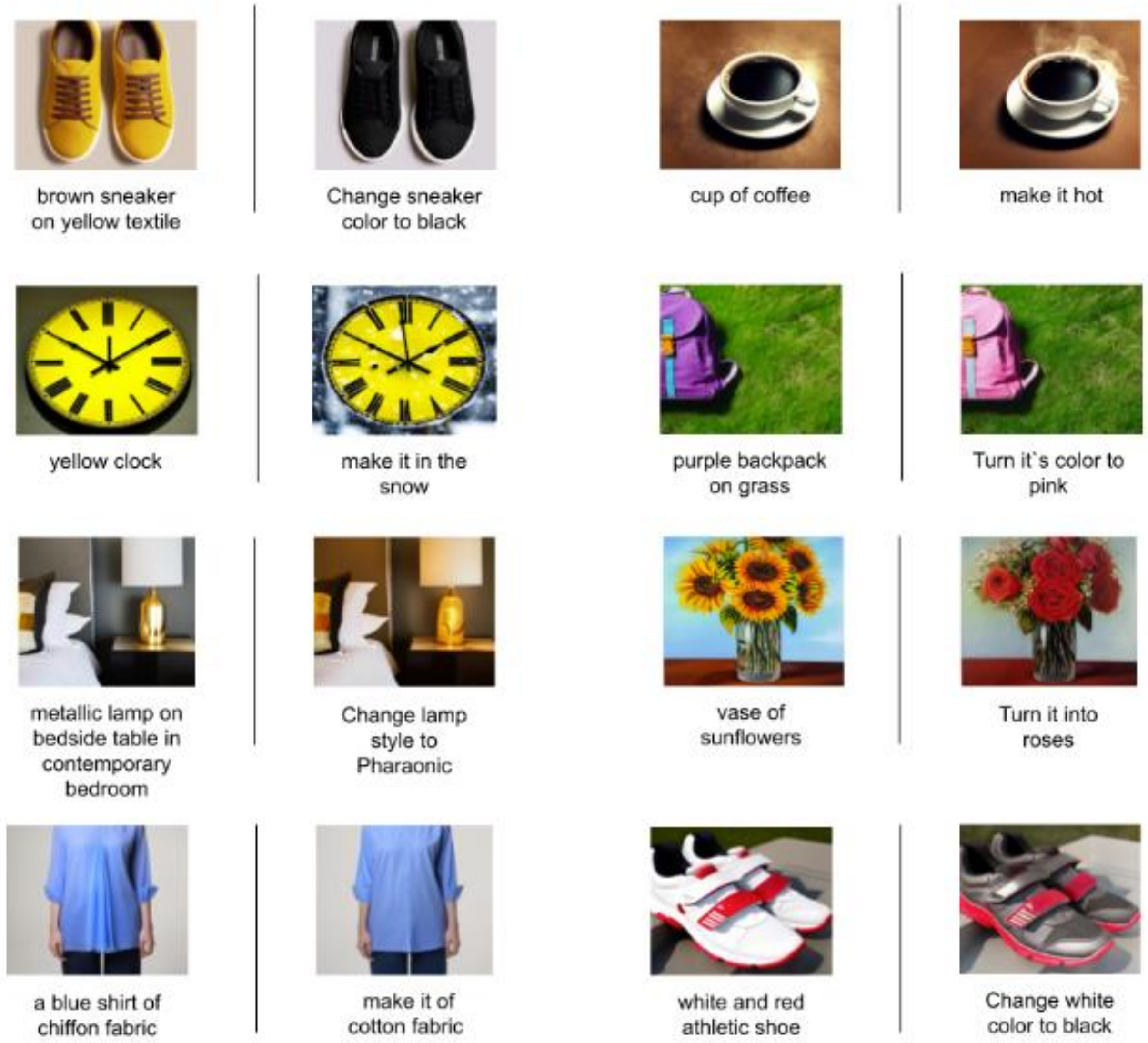

brown sneaker on yellow textile
Change sneaker color to black
cup of coffee
make it hot
yellow clock
make it in the snow
purple backpack on grass
Turn it`s color to pink
metallic lamp on bedside table in contemporary bedroom
Change lamp style to Pharaonic
vase of sunflowers
Turn it into roses
a blue shirt of chiffon fabric
make it of cotton fabric
white and red athletic shoe
Change white color to black

Figure 4.2: Generated Dataset Samples

Table 4.1 shows a summary of our generated dataset.

	Manual Dataset	**Automated Dataset**
Number of categories	50	10
Number of different edits	50	25
Total number of rows	50	3000
Number of images for each row	6	6-10
Final size	~300	~12 000

Table 4.1: Dataset Summarization

4.2 Experiments and Results

In this section, the various experiments and results applied throughout the work on our project will be presented.

- Trial 1:

  We initially worked on the Dreambooth model [11], but it uses inputs different from our architecture. Specifically, Dreambooth requires 3-5 input images, while we aimed to use only one input image. Afterward, we tried different models [13, 14, 16], but all these models required substantial memory and GPU resources, so we tried to leverage the resources provided by Colab and Kaggle.

- Trial 2:

  In this experiment, we tried using the Sine model [14]. As shown in Figure 4.3, the sine model consists of two main parts in its architecture. The first part is training the text-to-image diffusion model on the dataset, and the second part is fine-tuning the text-to-image diffusion model on the input image. The output image is generated by using a linear combination function for both parts.

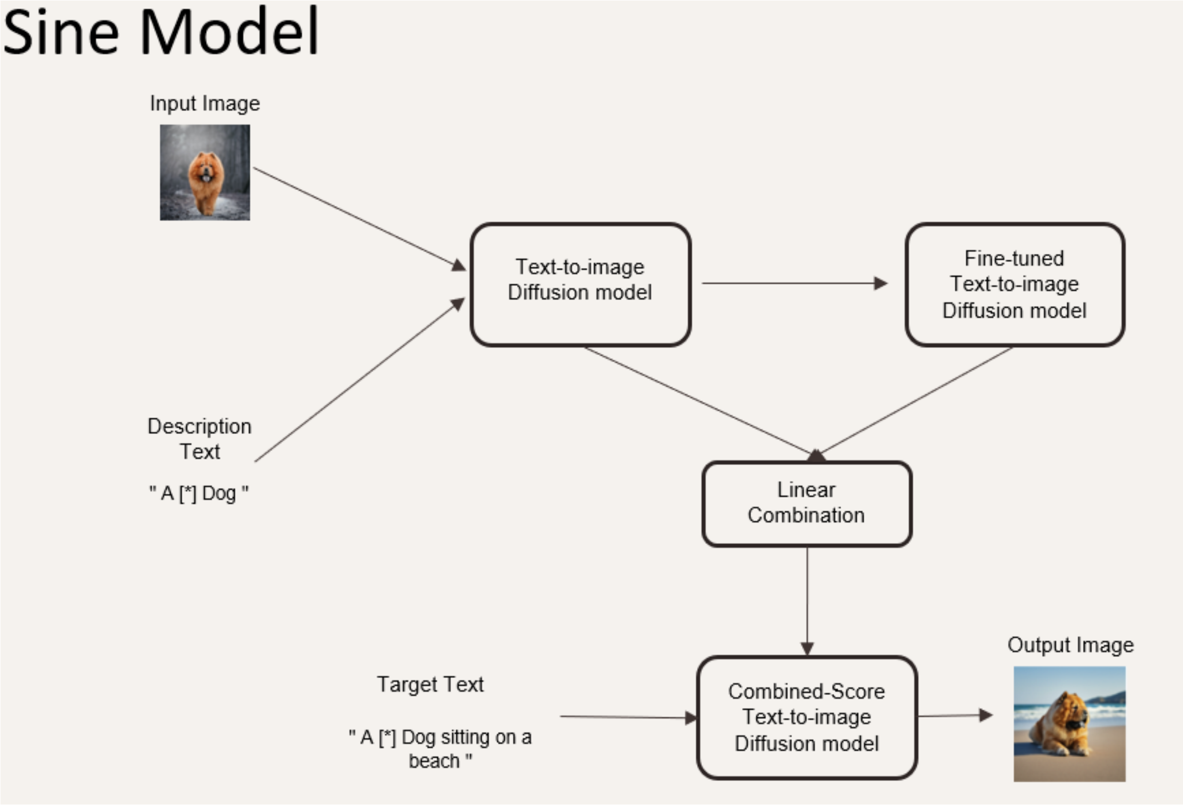
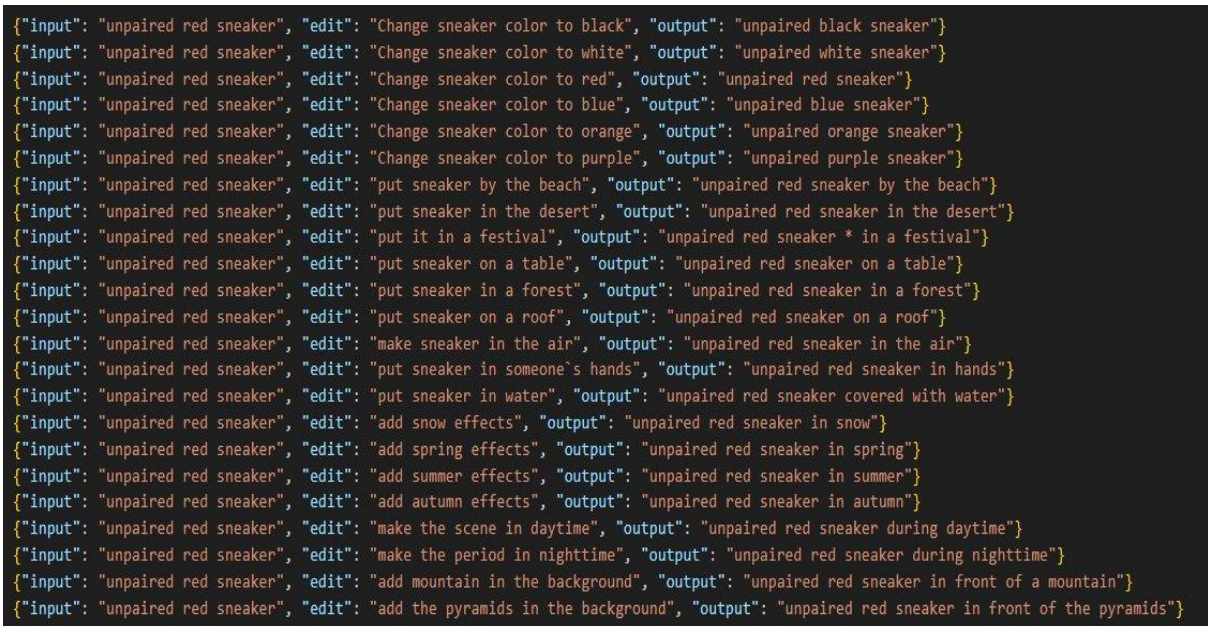

Sine Model
Input Image
Text-to-image Diffusion model
Fine-tuned Text-to-image Diffusion model
Description Text
" A [*] Dog "
Linear Combination
Output Image
Target Text
" A [*] Dog sitting on a beach "
Combined-Score Text-to-image Diffusion model

Figure 4.3: Sine model architecture with example.

The SINE model [14] is designed for single image editing tasks and adopts an approach similar to Dreambooth. It begins by taking the input image and extracting the object class. The input text is then modified by replacing the label or class name with a unique identifier and the class name. For instance, if the description text is "A dog," it becomes "a [*] dog." By combining this modified description text "a [*] dog" with the input text "A dog on a beach," the target text becomes "A [*] dog on a beach." This target text, along with the input image, is used to fine-tune the stable diffusion model. The architecture of the SINE model involves training a text-to-image diffusion model on a dataset, followed by fine-tuning the model on the specific input image. The final output is generated by combining the results from both the fine-tuned diffusion model and the original diffusion model using a linear combination function. This approach helps produce images that align with the text description while preserving the input image's characteristics. However, the fine-tuned text-to-image diffusion model requires substantial resources, making it challenging to fully utilize the architecture as depicted in Figure 4.3. Despite this, we were able to generate output images by omitting the fine-tuning part, though the results were less accurate and might differ from the input image.

Due to resource limitations that prevented the SINE model architecture from preserving object features, we transitioned to the InstructPix2Pix model [16]. This model was chosen for its efficiency in performance and memory usage. However, it required the creation of a product-based dataset specifically designed to fine-tune the InstructPix2Pix model which led to trial 3.

- Trial 3:
  Initially, the textual dataset generation was attempted using a large language model (LLM) such as GPT-3. However, due to the high associated costs, we adopted an alternative approach: a hardcoded text synthesizer. This synthesizer accepts a JSON file containing randomly generated input texts and maps these texts to various edit instructions related to attributes such as color, weather, material, style, and seasons. The process outputs a JSON file that includes the input text, the corresponding edit instruction, and the resultant output text. Despite being considered a naive approach for textual data generation, the hardcoded text synthesizer produced high-quality text prompts, proving successful in creating the required dataset. Figure 4.4 shows an example of a JSON file resulting from the hardcoded text synthesizer.



```
{"input": "unpaired red sneaker", "edit": "Change sneaker color to black", "output": "unpaired black sneaker"}
{"input": "unpaired red sneaker", "edit": "Change sneaker color to white", "output": "unpaired white sneaker"}
{"input": "unpaired red sneaker", "edit": "Change sneaker color to red", "output": "unpaired red sneaker"}
{"input": "unpaired red sneaker", "edit": "Change sneaker color to blue", "output": "unpaired blue sneaker"}
{"input": "unpaired red sneaker", "edit": "Change sneaker color to orange", "output": "unpaired orange sneaker"}
{"input": "unpaired red sneaker", "edit": "Change sneaker color to purple", "output": "unpaired purple sneaker"}
{"input": "unpaired red sneaker", "edit": "put sneaker by the beach", "output": "unpaired red sneaker by the beach"}
{"input": "unpaired red sneaker", "edit": "put sneaker in the desert", "output": "unpaired red sneaker in the desert"}
{"input": "unpaired red sneaker", "edit": "put it in a festival", "output": "unpaired red sneaker * in a festival"}
{"input": "unpaired red sneaker", "edit": "put sneaker on a table", "output": "unpaired red sneaker on a table"}
{"input": "unpaired red sneaker", "edit": "put sneaker in a forest", "output": "unpaired red sneaker in a forest"}
{"input": "unpaired red sneaker", "edit": "put sneaker on a roof", "output": "unpaired red sneaker on a roof"}
{"input": "unpaired red sneaker", "edit": "make sneaker in the air", "output": "unpaired red sneaker in the air"}
{"input": "unpaired red sneaker", "edit": "put sneaker in someone`s hands", "output": "unpaired red sneaker in hands"}
{"input": "unpaired red sneaker", "edit": "put sneaker in water", "output": "unpaired red sneaker covered with water"}
{"input": "unpaired red sneaker", "edit": "add snow effects", "output": "unpaired red sneaker in snow"}
{"input": "unpaired red sneaker", "edit": "add spring effects", "output": "unpaired red sneaker in spring"}
{"input": "unpaired red sneaker", "edit": "add summer effects", "output": "unpaired red sneaker in summer"}
{"input": "unpaired red sneaker", "edit": "add autumn effects", "output": "unpaired red sneaker in autumn"}
{"input": "unpaired red sneaker", "edit": "make the scene in daytime", "output": "unpaired red sneaker during daytime"}
{"input": "unpaired red sneaker", "edit": "make the period in nighttime", "output": "unpaired red sneaker during nighttime"}
{"input": "unpaired red sneaker", "edit": "add mountain in the background", "output": "unpaired red sneaker in front of a mountain"}
{"input": "unpaired red sneaker", "edit": "add the pyramids in the background", "output": "unpaired red sneaker in front of the pyramids"}
```

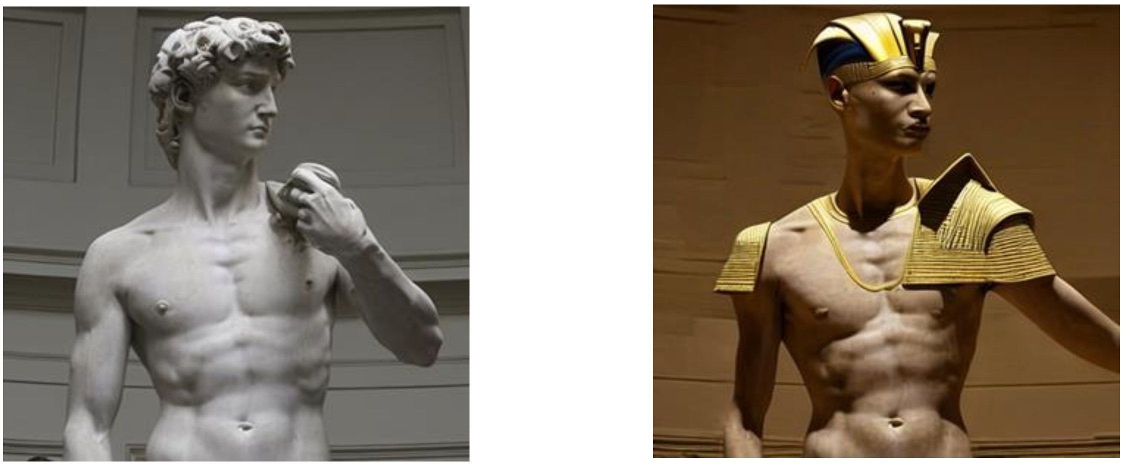
Figure 4.4: Samples from the automated textual Dataset.

- Trial 4:

  The training process for InstructPix2Pix requires substantial memory and GPU resources. According to its authors [16], the minimum resources needed include 64 GB of VRAM. To address this, we attempted to minimize the parameters and partition the dataset during the training process to enable fine-tuning. However, the resources provided by Kaggle were still insufficient.

- Trial 5:

  This trial involved testing various platforms offering free resources, including Google Cloud and Microsoft Azure, which supported only CPUs, all of which proved unsuitable except Kaggle. Consequently, Kaggle was utilized for both dataset generation and model deployment. Figure 4.5 illustrates a representative instance of the model being employed on Kaggle.



Turn him into a pharaoh.

Figure 4.5: Example on the model output

Additionally, we submitted a proposal to The Supercomputing Facility, Bibliotheca Alexandrina, requesting access to their GPUs. Our proposal was accepted; however, we are currently awaiting access to their supercomputers.

- Trial 6:

  The model itself is deployed on Kaggle, but the user interface is on Hugging Face. The process of generating the output image follows the steps shown in Figure 4.6. These steps can be summarized as follows:

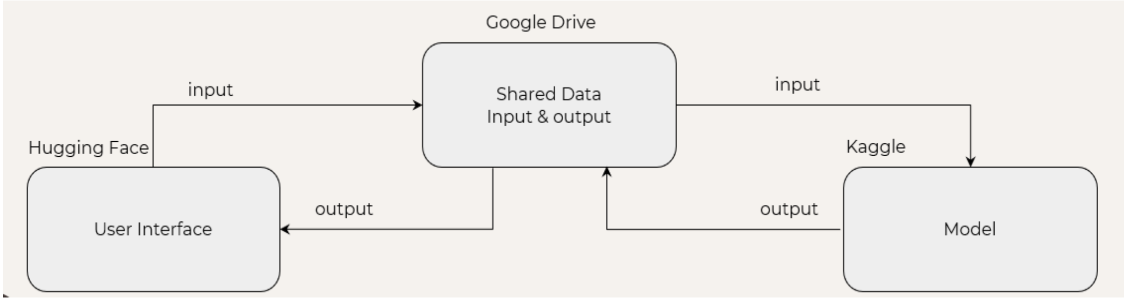


Figure 4.6: Deployment.

a. The user enters the input on Hugging Face, then the input is uploaded to Google Drive using the Google Drive API.
b. A trigger is then called using Kaggle-API to inform the notebook on Kaggle to start running. Once the notebook starts running, it first installs the essential dependencies, which takes approximately 20 minutes. It then retrieves the input from Google Drive, runs the model to generate the output image, and uploads the image to Google Drive.
c. On Hugging Face, we continuously check if the notebook on Kaggle has been completed and run successfully. If so, we retrieve the output image from Google Drive and display it to the user.

4.3 Testing

In this section, we explain the testing phase, which consists of two parts: model testing and deployment testing.

1. Model Testing:

SINE model:

Figure 4.7 shows the difference between using Sine [14] with the process of fine-tuning the diffusion model on the input and without using it, as we can see the output image differs from the input image.

As show

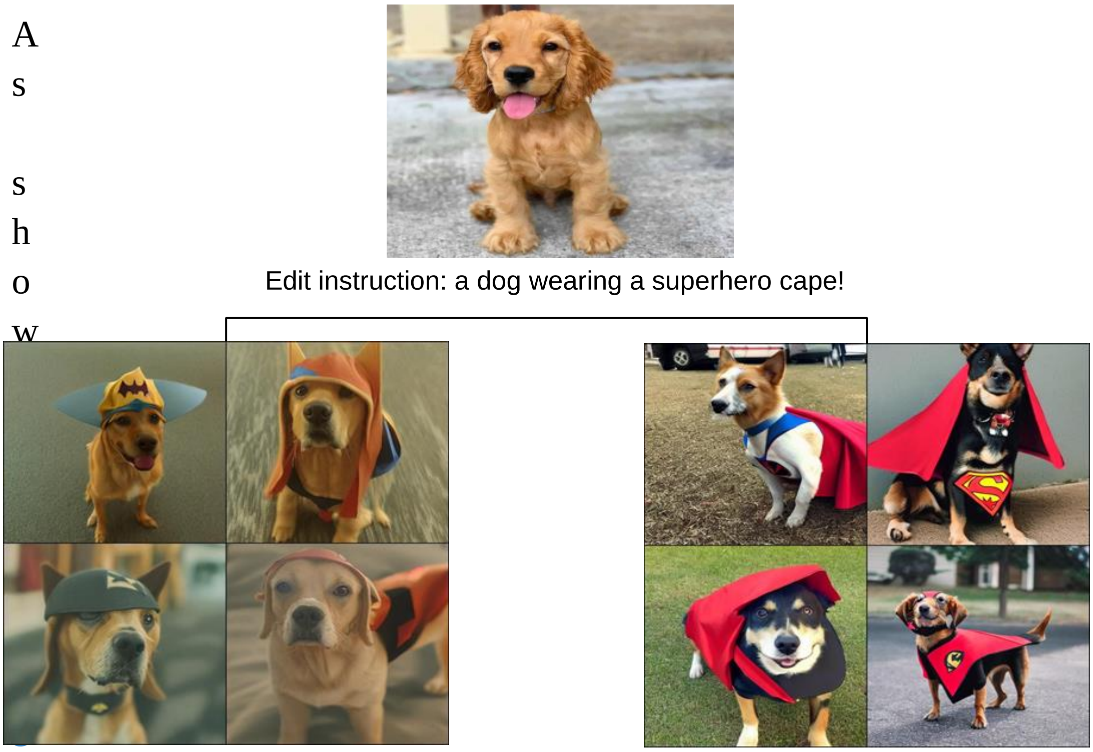
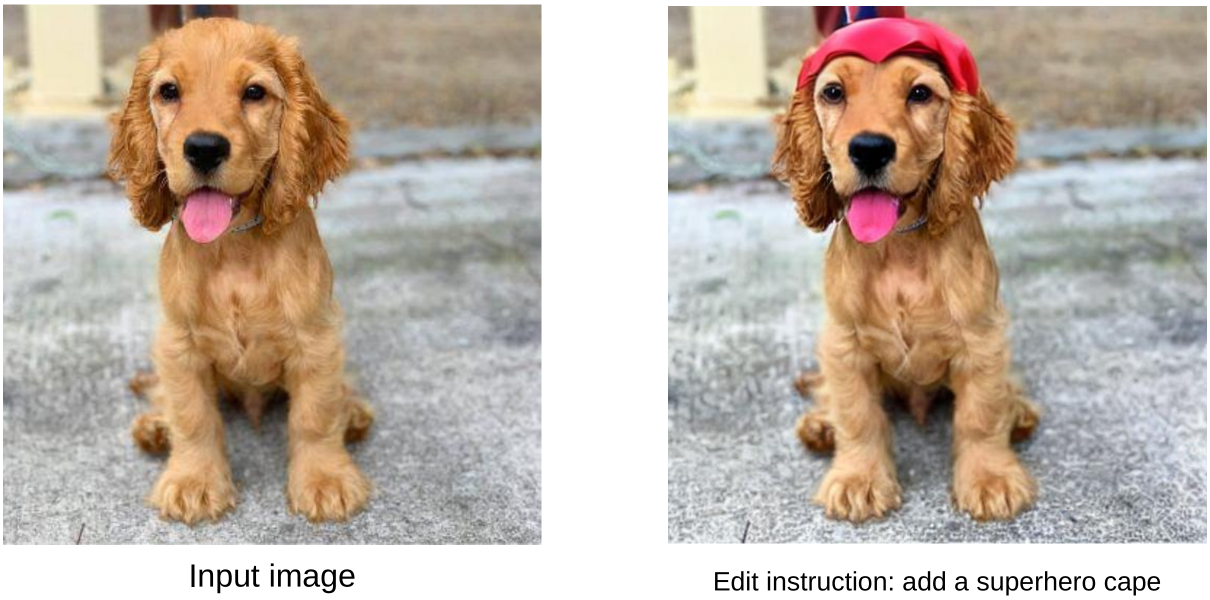


Figure 4.7: The difference with/without fine-tuning process in Sine.

As shown in Figure 4.7 the model successfully executed the edit instructions in both outputs. However, without fine-tuning, the model overlooks the importance of maintaining the input image's characteristics, resulting in the object in the figure differing from the original input image.

InstructPix2Pix:

This model requires fewer resources to generate the output image and demonstrates superior alignment between the output image and the target text compared to the SINE model. Figure 4.8 illustrates a representative instance of the model output.

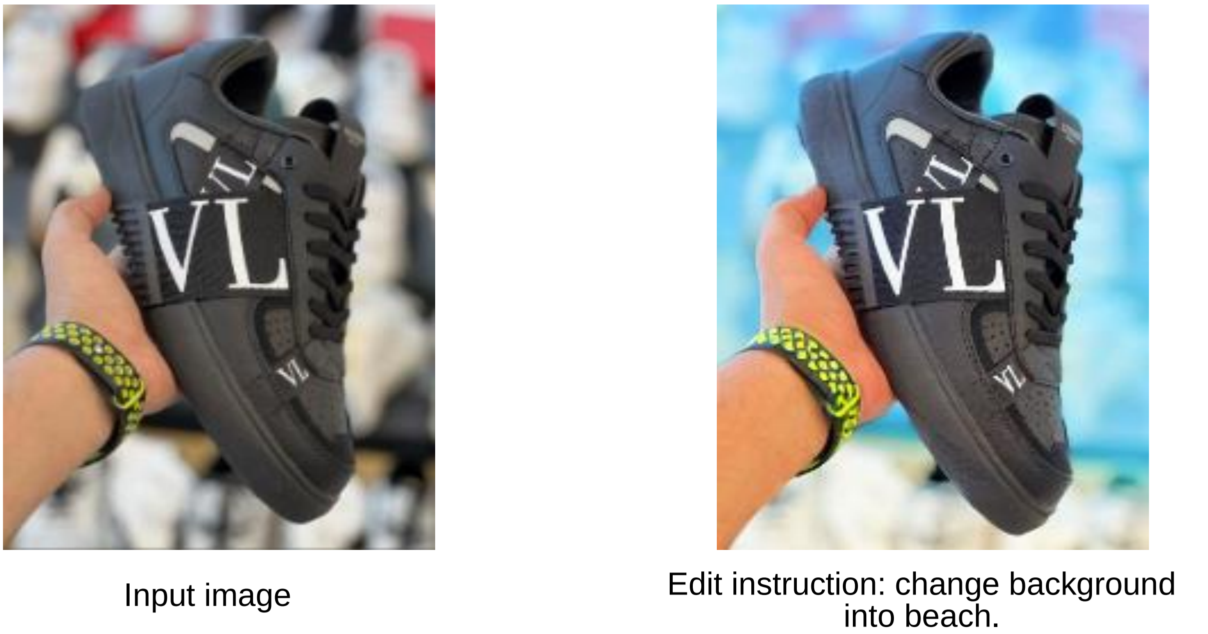


Input image

Edit instruction: add a superhero cape

Figure 4.8: Example on InstructPix2Pix2 model.

Moreover, We encountered two notable limitations: it may not perform optimally with real images, and it necessitates precise editing instructions for effective operation.



Input image

Edit instruction: change background into beach.

Figure 4.9: Example on a real, blurry image.

as shown in Figure 4.9 editing a real image is less accurate due to image quality, blurring, and other aspects.

2. Deployment Testing:

1. Overview

This phase involved rigorous testing of the model's integration, performance, and scalability on cloud platforms.

2. Setup and Environment

The deployment utilized Kaggle for model execution and Hugging Face for the user interface, establishing a robust and scalable infrastructure. Key components included:

- Compute Resources: Utilization of Kaggle's GPU resources for executing the model.
- Storage: Google Drive for storing input images and generated outputs.
- Network Configuration: Ensuring efficient data flow between Kaggle, Hugging Face, and Google Drive.

3. Workflow Integration

The deployment workflow integrates Hugging Face and Google Drive with the following steps:

- User Input: Users input data on the Hugging Face interface, which is then uploaded to Google Drive via the Google Drive API.
- Kaggle Execution: The Kaggle API triggers notebook execution, installs dependencies, processes the input data, and generates the output image.
- Output Retrieval: The Hugging Face interface retrieves the generated image from Google Drive and displays it to the user.

4. Execution Constraints

We encountered specific challenges due to the constraints of running the model on Kaggle:

- Single Notebook Execution: Only one notebook can run at a time on Kaggle, preventing parallel usage for multiple users.

- Dependencies Installation Time: Each notebook execution creates a new version of the notebook, which requires installing dependencies. This process takes approximately 20 minutes.

These constraints necessitated optimization of the deployment process to support parallel usage and ensure efficient execution.

5. Resource Management

The InstructPix2Pix model demands substantial memory and GPU resources. We explored various platforms, including Google Cloud and Microsoft Azure, but they only offered free CPUs. We chose Kaggle for its suitability in both dataset generation and model deployment.

6. System Reliability and Maintenance

To ensure the system works as expected, we implemented unit testing and added helper functions to manage the image path correctly. Because the model is deployed on Kaggle, we needed to check whether the notebook had finished its execution. We implemented a mechanism that checks the notebook status every 120 seconds. If there is an error during execution, it throws an exception, ensuring reliable and efficient operation.

7. Enhancements

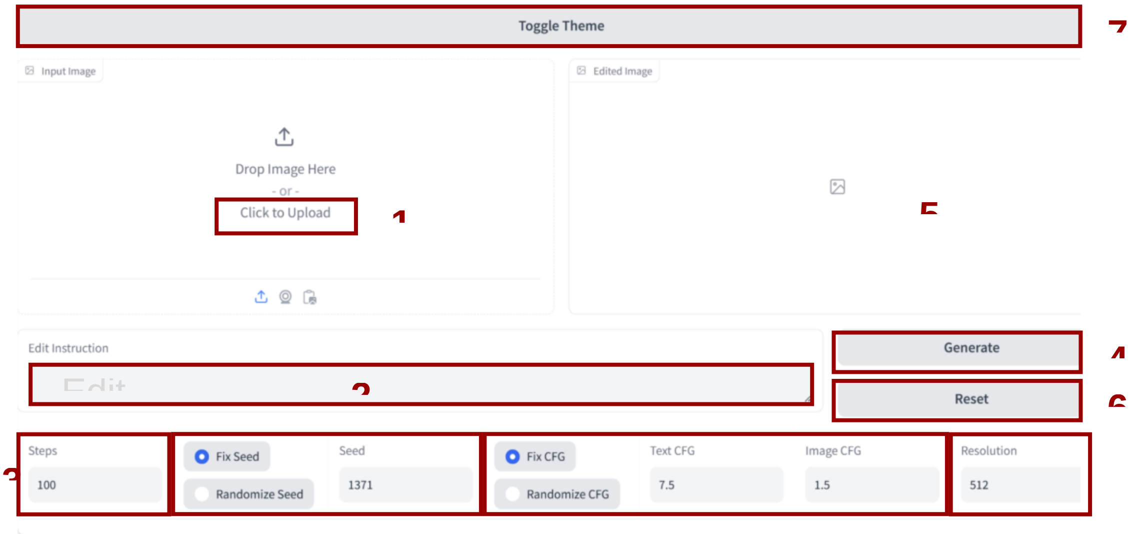
To address the execution time constraints and parallel usage challenges associated with using Kaggle as a model host, we have proposed utilizing The Supercomputing Facility at Bibliotheca Alexandrina. This proposal has been accepted, and we are awaiting access to further enhance our deployment capabilities.

5- User Manual

This chapter is a walkthrough of the whole website.

For the user to get the generated image, they will need to add the original image and the required edit text.

Figure 5.1 shows a panoramic overview of our website’s main interface.



Steps: 100

● Fix Seed
○ Randomize Seed

Seed: 1371

● Fix CFG
○ Randomize CFG

Text CFG: 7.5

Image CFG: 1.5

Resolution: 512

Figure 5.1: Abstract for all steps.

- As shown in Figure 5.1, to generate a new image, first press on **click to upload (1)** in the input box and choose the desired photo from the gallery then write the **edit instruction (2)** you want to apply then **adjust the parameters if needed (3)** and finally press on **generate button (4)** and the generated photo will appear in **edit box (5)**.
- To reset the website, add another photo, and edit instruction press on **reset button (6)**.
- To change the theme of the website press on **toggle theme button (7),** if it is on light mode, it will be on dark mode, and vice versa.

- **Description of each parameter**

  If you are not a technical user, you can leave all the parameters on default without any change.

1. **Steps**

   Number of training steps.

2. **Seed**

3. CFG (Classifier free guidance)

A. Text CFG

The effect of the text on the generated image, as it increases the change in the generated image increases.

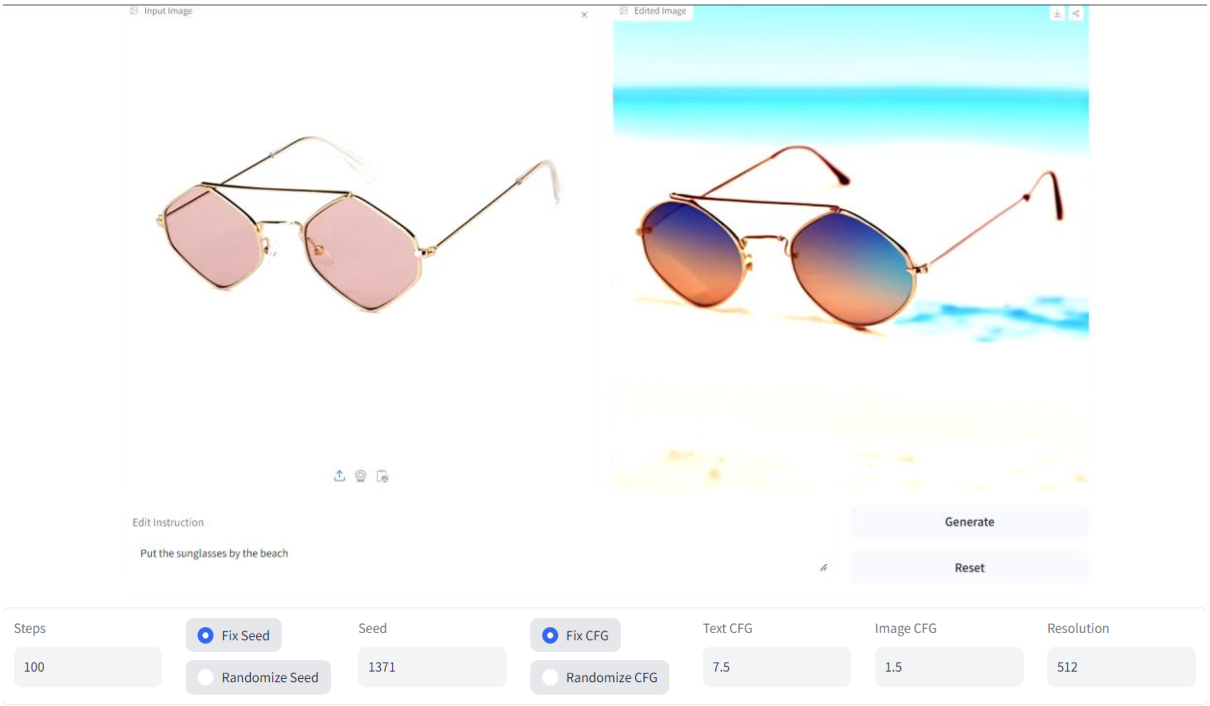
B. Image CFG

How much the generated image will be similar to the input image, as it increases the generated image will be similar to the input image.

4. Resolution

The quality and resolution of the generated image.

Figure 5.2 shows an example of a full run process:



Edit Instruction

Put the sunglasses by the beach

Steps

100

● Fix Seed

○ Randomize Seed

Seed

1371

● Fix CFG

○ Randomize CFG

Text CFG

7.5

Image CFG

1.5

Resolution

512

Figure 5.2: Example.

6- Conclusion and Future Work

6.1 Conclusion

In summary, this project aimed to develop a system for text-driven image-to-image generation, leveraging advanced deep learning models and algorithms to automate the process of updating images based on text inputs. The project involved several key components:

System Development: We created a user-friendly interface that allows users to upload an image, provide text instructions for modifications, and generate updated images. The system utilizes a text-to-image diffusion model. The process is designed to maintain high visual quality while accurately applying the textual modifications. The user interface was hosted on Hugging Face.

Dataset Generation: A significant part of the project was generating a suitable dataset. Initially, we explored using large language models like GPT-3 for text prompt generation but opted for a more resource-efficient approach using a hardcoded text synthesizer. The generated text prompts were high quality and suitable for creating the necessary dataset. The dataset generation and model deployment were facilitated using Kaggle. Figure 4.2 illustrates various input images along with their text descriptions, the applied edit instructions, and the resulting output images.

We uploaded our generated dataset as a hugging face dataset, Text-Driven-I2I-Generation-Dataset [19].

Experiments and Results: Several trials evaluated the system's performance. The initial sine model [14] required extensive resources and was partially excluded, resulting in generally accurate but imperfect images as shown in Figure 4.7. Due to resource limitations, the approach switched to the InstructPix2Pix model [16], necessitating a new product-based dataset for fine-tuning.

Though the initial model was lightweight, the fine-tuned text-to-image diffusion model required extensive resources. Ultimately, InstructPix2Pix model [16] was adopted for its efficiency in performance and memory usage.

Overall, this project provides a robust solution for content creators, educators, marketers, and social media users, enabling the efficient creation of customized and engaging visual content. By automating image modifications based on text inputs, the system significantly enhances productivity and creativity across various domains.

6.2 Future Work

Prompt Engineering: Enhance the natural language processing component to better interpret complex and nuanced text prompts. Incorporate context-aware models and develop techniques to give users explicit control over object positioning and attributes within the generated images, ensuring more precise and desired outputs.

Image to Image Enhancement: Implement super-resolution models to enhance the quality and resolution of generated images. Integrate denoising algorithms to clean up artifacts and improve visual quality. Allow users to specify style and attribute enhancements in their text prompts for more personalized outputs. Additionally, enable object position swapping capabilities to allow users to rearrange and modify the position of objects within the generated images.

Deployment: To enable multi-user access for our model deployed on Hugging Face, we propose integrating load balancing to distribute requests across multiple instances. Additionally, deploying the model on a scalable cloud infrastructure such as AWS, Google Cloud, or Azure will allow for autoscaling based on demand, ensuring seamless multi-user access.

References

1. B. Li, X. Qi, P. H. S. Torr, and T. Lukasiewicz "Image-to-Image Translation with Text Guidance" in University of Oxford, 2020.
2. F. Bao, S. Nie, K. Xue, Y. Cao, C. Li, H. Su, and J. Zhu "All are Worth Words: A ViT Backbone for Diffusion Models" in Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Center, 2022.
3. E. Segalis, D. Valevski, D. Lumen, Y. Matias, and Y. Leviathan "A Picture is Worth a Thousand Words: Principled Recaptioning Improves Image Generation", 2023.
4. S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee "Generative Adversarial Text to Image Synthesis" in Max Planck Institute for Informatics, Saarbrucken, Germany, 2016.
5. C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. Fleet, and M. Norouzi "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding" in Google Research, Brain Team Toronto, Ontario, Canada, 2022.
6. X. Wang, S. Fu, Q. Huang, W. He, and H. Jiang "MS-Diffusion: Multi-subject Zero-shot Image Personalization with Layout Guidance", 2024.
7. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer "High-Resolution Image Synthesis with Latent Diffusion Models" in Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany, 2021.
8. P. Isola, J. Z. Tinghui, Z. Alexei, and A. Efros "Image-to-Image Translation with Conditional Adversarial Networks" in Berkeley AI Research (BAIR) Laboratory, UC Berkeley, 2017.
9. S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M. Yang, and L. Shao "Learning Enriched Features for Real Image Restoration and Enhancement" in Inception Institute of Artificial Intelligence, UAE, Mohamed bin Zayed University of Artificial Intelligence, UAE, University of California, Merced, USA, Google Research, 2020.
10. J. Zhu, T. Park, P. Isola, A. and A. Efros "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" in Berkeley AI Research (BAIR) laboratory, UC Berkeley, 2017.
11. N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman "DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation" in Boston University, 2022.
12. B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani "Imagic: Text-Based Real Image Editing with Diffusion Models" in Weizmann Institute of Science, 2022.
13. N. Tumanyan, M. Geyer, S. Bagon, and T. Dekel "Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation" in Weizmann Institute of Science, 2022.
14. Z. Zhang, L. Han, A. Ghosh, D. Metaxas, and J. Ren "SINE: SINgle Image Editing with Text-to-Image Diffusion Models" in Rutgers University, Snap Inc, 2022.
15. H. Chen, Y. Zhang, S. Wu, X. Wang, X. Duan, Y. Zhou, and W. Zhu "DisenBooth: Identity-Preserving Disentangled Tuning for Subject-Driven Text-to-Image Generation" in Department of Computer Science and Technology, Tsinghua University, Beijing National Research Center for Information Science and Technology, Lanzhou University, 2023.
16. T. Brooks, A. Holynski, and A. A. Efros "InstructPix2Pix: Learning to Follow Image Editing Instructions" in University of California, Berkeley, 2022.

17. A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or "Prompt-to-Prompt Image Editing with Cross Attention Control" in The Blavatnik school of computer science tel aviv university, 2022.
18. P. Dhariwal, and A. Nichol "Diffusion Models Beat GANs on Image Synthesis", 2021.
19. Hugging Face, "Text-Driven-I2I-Generation-Dataset." Available: https://huggingface.co/datasets/Graduation-Proect-Team/Text-Driven-I2I-Generation-Dataset.
20. Hugging Face, "Text_Driven_Image_to_Image_Generation" Available: https://huggingface.co/spaces/Graduation-Proect-Team/Text_Driven_Image_to_Image_Generation

ملخص

يركز هذا المشروع على إنشاء نموذج يقوم بتحديث الصور بناءً على التعليمات النصية، وهو أمر ضروري لوسائل التواصل الاجتماعي والإعلانات وإنشاء المحتوى. يمكن للمستخدمين تحميل صورة وتقديم تعليمات نصية للتغييرات، والتي يمكن أن تشمل إضافة تعليقات توضيحية أو تعديل العناصر أو إضافة معلومات بصرية جديدة. يقوم النظام بمعالجة هذه التعليمات لتحديث الصورة مع الحفاظ على جودة بصرية عالية.تشمل الميزات الرئيسية واجهة سهلة الاستخدام، وخوارزميات متقدمة لفهم وتطبيق التعليمات النصية، وإمكانية تغيير العناصر البصرية حسب الحاجة. في التجارب مع نماذج متقدمة لتحويل النص إلى صورة، تم مواجهة عقبة كبيرة: الحاجة إلى موارد ذاكرة ووحدة معالجة رسومات كبيرة. ومع ذلك، تميز نموذجان بالكفاءة وخفة الوزن.النموذج الأول الذي تم تجربته هو نموذج SINE والذي يتكون من جزأين رئيسيين: تدريب نموذج الانتشار من النص إلى الصورة على مجموعة البيانات وضبطه على الصورة المدخلة. يتم إنشاء الصورة الناتجة باستخدام دالة تركيب خطي للجزأين. على الرغم من خفة وزنه، إلا أن النموذج المضبوط للانتشار من النص إلى الصورة لا يزال يتطلب موارد كبيرة لتشغيله.في النهاية، تم استخدام نموذج InstructPix2Pix والذي أثبت كفاءته في الأداء واستخدام الذاكرة. كانت إحدى التحديات البارزة هي نقص مجموعات البيانات المخصصة في مجال تحويل النص إلى صورة. لمعالجة ذلك، تم التركيز على إنشاء مجموعة بيانات مخصصة، واختيار "المنتجات" كالمجال المطلوب. شمل عملية إنشاء مجموعة البيانات عمليتين فرعيتين: توليد البيانات النصية وتوليد البيانات الصورية.عادةً ما تستخدم الأعمال المشابهة نموذج لغة كبير (LLM) مثل GPT-3 لتوليد البيانات النصية. ولكن، بسبب التكاليف العالية المرتبطة باستخدامه، تم اختيار نهج أبسط: مولد نصوص مبرمج مسبقًا. على الرغم من أن هذه الطريقة تعتبر بدائية، إلا أنها قدمت تعليمات نصية عالية الجودة تم استخدامها بنجاح لإنشاء مجموعة البيانات اللازمة. أظهرت الاختبارات أن النظام يعمل بشكل موثوق، منتجًا تحديثات واضحة ودقيقة بناءً على أنواع مختلفة من الصور والتعليمات النصية.في الختام، يوفر هذا المشروع طريقة بسيطة وفعالة لتحديث الصور باستخدام التعليمات النصية ويقدم مجموعة بيانات متخصصة للمنتجات، وهي نادرة في هذا المجال. يستفيد منه منشئو المحتوى، والمعلمون، والمسوقون، ومستخدمي وسائل التواصل الاجتماعي مما يساعدهم فى انشاء محتوى بصري مخصص وجذاب