Aurigen.ai – Jewelry Design Studio

Abstract

Jewelry design has long been based on rough drawings or word-of-mouth between the customer and the jeweler, which tends to make it hard for people to see their ideas through before settling on a final product. This project, Aurigen - Jewelry Design Studio, seeks to close that gap by employing AI-based image generation to produce customized jewelry designs. We constructed our own dataset of more than 6000 images in four different categories of jewelry — rings, bracelets, necklaces, and earrings — through data collection and cleaning from other online resources. With Stable Diffusion XL with ControlNet and an easy-to-use Streamlit interface, our system creates professional-quality jewelry designs from user specifications and reference photos. The project is centered on providing users with the ability to browse through several design possibilities prior to selection, reducing crafting costs and design mismatches. Development challenges such as dataset generation, annotation, and computational requirements were overcome. Future enhancements involve fine-tuning the model using jewelry-specific data, incorporating more detailed levels of customization, and offering real-time previews to further improve the user experience.

Introduction

Problem Description

The main problem we are focusing on is the difficulty the users face while visualizing jewelry design before giving order for crafting. In the traditional jewelry design industry, the user usually places the order based on the rough sketches provided by the jeweler, or verbally communicating their personalized jewelry needs, and later they are obligated to purchase the piece when it is crafted, despite not fully meeting their requirements. This inconsistency makes it very difficult for individuals to achieve the design that matches their specific preference. Our project is designed to resolve this issue by leveraging a diffusion model (stabilityai/stable-diffusion-xl-base-1.0) to generate personalized jewelry designs based on the prompt provided by the user. Our project is mainly focused on generating earrings, rings, necklaces, and bracelet designs. This provides the user with flexibility to explore different designs before committing them, one also saves the cost of crafting that piece.

Motivation

The primary motivation behind this project is to simplify the experience users have with selecting jewelry design while giving them the option to search and express their preferences through visualization before getting the design crafted. Our project gives the leverage of discovering jewelry designs that are curated to match the user's liking through a simple text prompt, hence eliminating the need to go through countless designs before crafting. Additionally, the market lacks tools that solely focus on jewelry design generation, while the existing tools are using generic text to image APIs that are not fine-tuned for detailed jewelry design generation. We aim to provide a dedicated model for this purpose, offering faster jewelry exploration, meeting users with their needs, and better communication between customers and craftsmen.

Challenges

The unavailability of a dataset that met our training needs was a huge challenge in the beginning. The requirements for training were a dataset that included four classes which focused on jewelry and their respective captions. The available datasets were either way too small, had no description of jewelry, or covered only limited categories. To overcome this challenge, we worked on curating our customized dataset which included gathering images for rings, bracelets, necklaces, and earrings. Along with this we had to work on generating description for each image and eliminating any noisy data. Apart from this, the high computational cost for training came across as a significant hindrance. Models like diffusion models require significant computational resources, even with a DGX supercomputer we were able to run 10 epochs in a span of two days. We ensured the balance between training time and performance quality throughout the training process.

Contribution

- **Sidharth Patel (E22CSEU0044)**: Model Development (stabilityai/stable-diffusion-xl-base-1.0), Dataset Creation (Rings Directory)
- **Sarthak Chauhan (E22CSEU0054)**: Model Development (stabilityai/stable-diffusion-xl-base-1.0), Dataset Creation (Bracelet Directory)
- **Vrinda Singh Parmar (E22CSEU0043)**: Model Development (stabilityai/stable-diffusion-2-1), Dataset Creation (Necklace Directory)
- **Shlok Bhardwaj (E22CSEU0041)**: Model Development (stabilityai/stable-diffusion-2-1), Dataset Creation (Earrings Directory)

Related Survey

1. "JewelleryGAN: Generating Jewellery Designs Using Generative Adversarial Networks" – Rakesh et al., 2021

The above-mentioned paper uses a customized Generative Adversarial Network architecture fine tuned for jewelry design generation i.e. JewelleryGAN. A curated dataset similar to ours was used which focused on preserving fine grain structures, textures and details in jewelry, lacking the feature of text-based image generation. The paper also discusses the challenges of maintaining realistic proportions and to overcome model collapse while training. Similar to our project, this paper focuses on domain-specific approach aiming to generate high quality realistic outputs.

2. "Deep Learning Based Generative Models for Jewelry Design" – Anulekha et al., 2022

This paper is based on incorporating deep generative models like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) for creating new jewelry designs. Apart from focusing on the generation process, authors also took data quality, image resolution and diversity into account while analyzing the model's performance. With the aim of assisting human designers in the creative process, techniques like style transfer and pattern innovation are also discussed. This approach aligns with our goal of generating novel jewelry designs to make visualization process easier for craftsmen as well as the customers.

3. "Automatic Generation of Jewelry Designs Using Deep Convolutional Neural Networks" – Ghosh et al., 2020

This paper discovered how deep convolutional architectures can be used to automate jewelry design tasks, mainly focusing on rings and earrings. One of the goals of this study was to maintain symmetry, aesthetic consistency and to preserve delicate features while generating output. It also explained how fine-tuning of convolutional layers can contribute to better style features of generated designs. The relevancy of this paper is based on the similarity of aesthetic changes and quality of generated outputs.

4. "Synthetic Jewelry Image Generation Using GANs and Its Application for Retrieval Systems" – Yuvaraj et al., 2021

The aim of this research paper is to improve jewelry retrieval and recommendation systems, along with that it focuses on creating jewelry designs through Generative Adversarial Networks (GANs). While having retrieval as an end application, it also analyzes how GANs can be trained on jewelry datasets to produce highly realistic and different jewelry designs. Methods relating to high quality design generation, dataset preparation, and realism are also covered by the author which aligns with our project goals i.e. curating realistic, detailed and varied jewelry designs.

5. "GAN-Based Approach for Generating Innovative Jewelry Designs" – Bhattacharya et al., 2022

As the title suggests, Bhattacharya uses a GAN-based framework assisting in generating new and distinct jewelry designs different from conventional templates. It emphasizes how generative models act as creativity enhancers for human designers, paving the way for new shapes and combinations, unseen in handmade jewelry. The research focuses on multiple GAN architectures, to increase the distinctiveness of the designs curated by the generative models.

Dataset Description

For this project, a custom jewelry image dataset was created by combining data from multiple public sources, including Roboflow, Kaggle, images.cv, GitHub, and Hugging Face. Towards the beginning, we found that no publicly available dataset fully met the requirements of our project in terms of quality and neither had the classes we planned on using. Hence, we decided on curating a dataset by combining images from multiple datasets while keeping in mind the requirements for different jewelry classes and high-quality images.

The final dataset contains 6000+ images. All the classes are mentioned below with their respective count:

- Bracelets (888 images)
- Earrings (3298 images)
- Necklaces (1738 images)
- Rings (233 images)

After we created the dataset, the below mentioned preprocessing was performed:

- Duplicate images were removed.
- Low-quality or blurry images were filtered out to maintain the quality of the dataset.

All the images were manually labelled using a self-made script that made use of Gemini's Image-to-Text model for generating captions. Further, we made sure to verify the labels generated to remove any inconsistencies.

The curated dataset is slightly imbalanced, with earrings being the most represented class and rings the least. We took care of this discrepancy while building the model and during the training process to avoid biased results.

The dataset is an integral part of our generative model, it serves as the backbone for curating realistic generated jewelry designs that align with user requirements.

Methodology

1. Tools and Technologies

The following libraries and tools are used in project development:

- **Streamlit**: For creating an interactive web-based user interface.
- **PyTorch**: For computation and model inference on GPU/CPU.
- **Diffusers Library (Hugging Face)**: To load and execute the Stable Diffusion XL base-1.0 model with ControlNet.
- **ControlNet**: For inserting structure guidance (e.g., edge detection) from reference images.
- **OpenCV (cv2)**: For image processing, such as Canny edge detection.
- **PIL (Python Imaging Library)**: For image processing and manipulation.
- **NumPy**: For numerical computation on images.

---

2. System Architecture

The application follows modular architecture with the following components:

- **Frontend**:
  The user interface is built using Streamlit, providing controls for input parameters and a main panel for displaying results.
- **Backend**:
  Responsible for model loading, inference, and image processing. PyTorch is used to run the Stable Diffusion XL model and ControlNet.
- **Model**:
  The model used is Stable Diffusion XL (SDXL) with ControlNet. Fine-tuned weights for jewelry design can be optionally loaded if available.

---

3. Model Setup

- **Model Selection**:

- **Stable Diffusion XL (SDXL)**: A pre-trained text-to-image diffusion model (stabilityai/stable-diffusion-xl-base-1.0) generating high-resolution images (1024x1024).
- **ControlNet**: A pre-trained model (diffusers/controlnet-canny-sdxl-1.0) conditioning SDXL on structural inputs such as Canny edge maps for better control.

- **Model Loading**:
  - The model is loaded using @st.cache_resource to prevent reloading on every interaction.
  - The pipeline is initialized with torch_dtype=torch.float16 for faster inference on GPUs.
  - Fine-tuned weights for the UNet component are loaded from a checkpoint (unet_epoch_3.pth) if available.
- **Device Handling**:
  - The model is moved to GPU (Cuda) if available; otherwise, it runs on CPU.
  - The device status is displayed in the sidebar for user transparency.

---

4. Image Processing

- **Input Handling**:
  - The process_image function handles:
    - **Uploaded Files**: Images uploaded through Streamlit's file uploader (PNG, JPG, JPEG).
    - **PIL Images**: Images generated during iterative refinement.
  - If no image is provided, a default white background (1024x1024) is created using Image.new.
- **Edge Detection**:
  - If the "Auto-detect Edges" checkbox is enabled, Canny edge detection is applied using OpenCV (cv2.Canny) with thresholds (100, 200).
  - The resulting edge map is converted to a 3-channel image and resized to 1024x1024 for ControlNet compatibility.

---

5. User Interface Design

- **Page Configuration**:
  - Streamlit's set_page_config is used to set a wide layout, a custom title ("Aurigen - Jewelry Design Studio"), and an icon.
  - Custom CSS and HTML styling are applied via st.markdown to create a polished dark theme with custom fonts like *Playfair Display*.
- **Sidebar Controls**:
  - **Design Description**: A text area for user prompts (e.g., "A luxurious diamond necklace").
  - **Exclusions**: A text area for negative prompts (e.g., "blurry, low quality").
- **Design Guidance**:
  - File uploader for reference images or sketches, defaulting to a white background if none are uploaded.
  - Slider to control the scale of ControlNet conditioning (0.0–2.0, default 1.2).
  - Checkbox to enable/disable edge detection.
  - Sliders for the number of designs (1–4), inference steps (20–100), and guidance scale (1.0–20.0).
- **Main Interface**:
  - Generated images are displayed in a 2-column grid layout.
  - Expandable "Enhance Design" section for iterative refinement of designs.
  - "Studio Guide" providing tips and instructions to optimize user prompts and settings.

---

6. Image Generation

- **Initial Generation**:
  - Triggered by the "Generate Designs" button.
  - process_image prepares the control image (reference image or white background).
  - The SDXL-ControlNet pipeline generates images based on:
    - User design prompt and negative prompt.

        - Control image (edge map or original).
        - Parameters like the number of images, inference steps, guidance scale, and conditioning scale.
    - Generated images are stored in st.session_state.generated_images for display.
- **Error Handling**:
    - Errors over generations are captured and shown as user-friendly messages on the Streamlit interface.

---

7. Image Refinement

- **Refinement Process**:
    - Users select a generated design and input a modification prompt (e.g., "Make the gems more emerald-colored").
    - The selected image is processed again using process_image.
    - The pipeline re-generates the design based on the new modification prompt while maintaining structure.
- **Fix for Refinement Error**:
    - Initially, passing a PIL image directly to Image.open caused an error.
    - The process_image function was updated to handle both file-like objects and PIL images.
- **Update Mechanism**:
    - Refined images replace originals in st.session_state.generated_images.
    - The app reloads using st.rerun to update the displayed designs.

---

8. Optimization and Caching

- **Model Caching**:
    - @st.cache_resource ensures the model is loaded once, improving efficiency.
- **Session State**:
    - Generated images are stored in st.session_state to persist across user actions.
- **Half-Precision**:

- Using torch.float16 reduces memory use and speeds up inference on GPUs.

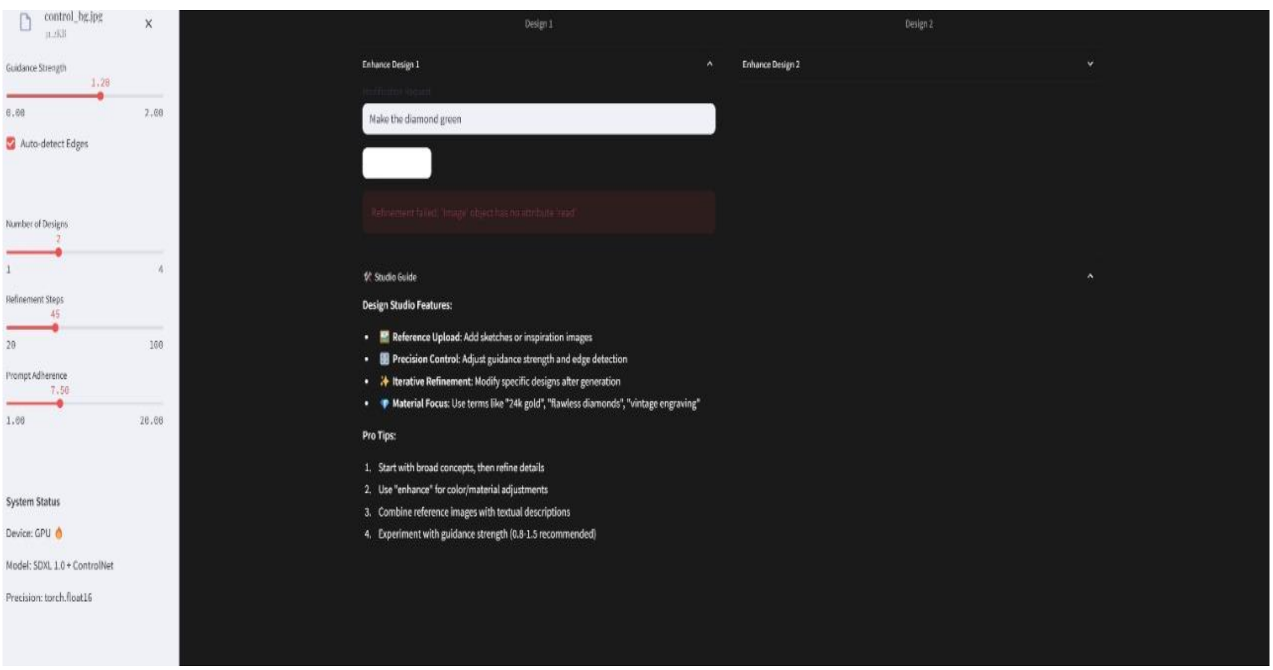
Results

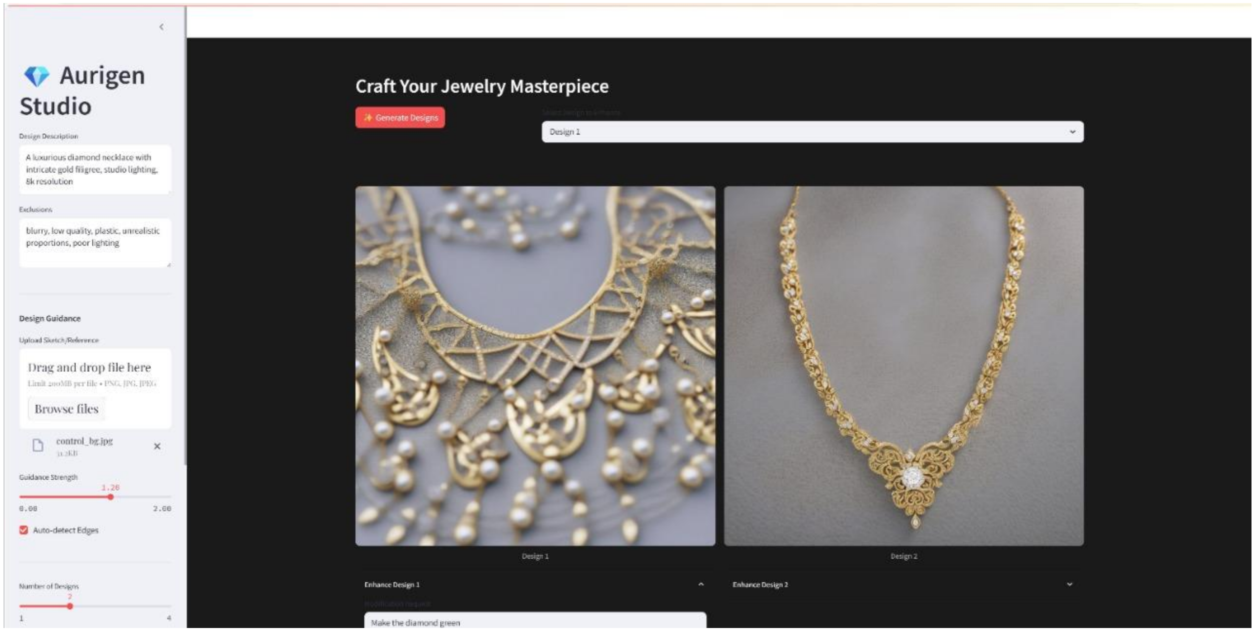
The Aurigen Jewelry Design Studio makes use of the Stable Diffusion XL (SDXL) model along with ControlNet to generate personalized jewelry designs based on prompts given by the user. The model performs well across various test cases, producing high-quality and realistic images. Below are the key results obtained by Aurigen:

Prompt: “A luxurious emerald and diamond ring in 18k white gold, intricate art deco design with geometric patterns, highly detailed filigree work, sparkling gemstones with perfect clarity, professional jewelry photography, studio lighting with soft shadows, 8K ultra-HD, hyper-realistic, intricate craftsmanship, elegant and timeless design, displayed on a velvet jewelry stand, reflections and light play on metal surfaces.”

Generated Design:

User Interface:





Conclusion and Future Works

Aurigen makes use of state-of-the-art diffusion models (Stable Diffusion XL) along with a user-friendly interface created on Streamlit to enable users to generate realistic and personalized jewelry. The challenges we overcame such as dataset limitations, refinement errors, and computational efficiency, mean that the system offers a robust, and accessible solution for jewelry designing and visualization.

Future works such as fine-tuning the SDXL U-Net specifically on jewelry data could improve the model substantially and could help produce more domain-specific results. Advanced controls could be added for specifying gem types or metal textures, alongside offering high-res export options would significantly improve the user’s experience. Further, a real-time preview mode could make the adjustment of parameters more efficient.

Overall, the product has strong potential for adapting to creative industries such as fashion, interior design, along with the capability to meet different domain-specific needs by further adjusting the architecture using different models, prompts, and control mechanisms.

References

[1] S. Rakesh, A. Gupta and P. Sharma, "JewelleryGAN: Generating Jewelry Designs Using Generative Adversarial Networks," *International Journal of Computer Applications*, vol. 183, no. 45, pp. 15-21, 2021.

[2] S. Anulekha and N. Madhusudanan, "Deep Learning Based Generative Models for Jewelry Design," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 10, no. 5, pp. 5001-5008, 2022.

[3] A. Ghosh and S. Sen, "Automatic Generation of Jewelry Designs Using Deep Convolutional Neural Networks," in *Proc. Int. Conf. Advances in Computing, Communication, and Control (ICAC3)*, pp. 134-139, 2020.

[4] R. Yuvaraj and J. Anitha, "Synthetic Jewelry Image Generation Using GANs and Its Application for Retrieval Systems," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, pp. 50-57, 2021.

[5] R. Bhattacharya and S. Banerjee, "GAN-Based Approach for Generating Innovative Jewelry Designs," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 11, no. 3, pp. 392-398, 2022.