Fashion-Specific Text-to-Image Generation

A Major-project Report submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications

by

SAGAR PURSWANI

(23419MCA046)

Department of Computer Science

Institute of Science

Banaras Hindu University, Varanasi

May 2025

CANDIDATE'S DECLARATION

I **Sagar Purswani** hereby certify that the work, which is being presented in the Major-project report, entitled **Fashion-Specific Text-to-Image Generation** , in partial fulfillment of the requirement for the award of the Degree of **Master of Computer Applications** and submitted to the institution is an authentic record of my own work carried out during the period January 2025 to May 2025 under the supervision of **Dr. Marisha.** I also cited the reference about the text(s) /figure(s) /table(s) /equation(s) from where they have been taken.

The matter presented in this Major-project as not been submitted elsewhere for the award of any other degree or diploma from any Institutions.

Sagar Purswani

Date: 24/05/2025

Signature of the Candidate

The Viva-Voce examination of Sagar Purswani, M.C.A. Student has been held on _____

Signature of the Supervisor	Signature of Examiner	Signature of Head of the Department

ABSTRACT

In recent years, generative AI has made significant advancements, particularly in the domain of text-to-image generation. However, the broad applicability of general-purpose models like **Stable Diffusion** poses a challenge for domain-specific needs such as fashion design, where outputs must be both relevant and safe for professional use. This project aims to solve this gap by developing a **fashion-specific text-to-image generation pipeline** that only accepts and processes fashion-related prompts and ensures safe, high-quality image outputs suitable for design prototyping and visualization.

The system is built on a **three-layer architecture**. The first layer uses prompt enhancement and validation mechanisms to preprocess user input and strictly filter out any non-fashion content using both semantic analysis and keyword-based rules. The second layer leverages a **Stable Diffusion model fine-tuned with a fashion-specific LoRA (Low-Rank Adaptation)** to generate high-resolution images that align with the designer's intent. The final layer consists of a postprocessing unit that applies **SwinIR for image enhancement** and **NudeNet for safety validation**, ensuring no inappropriate or irrelevant content is produced.

The entire pipeline is implemented using **open-source libraries** such as Hugging Face `diffusers`, `transformers`, `nudenet`, and `gradio`, and is deployed on GPU-enabled environment **Kaggle** using a user-friendly Gradio interface. The system returns an image only if the prompt is verified as fashion-related; otherwise, it responds with a message indicating the restriction.

This project not only showcases an innovative application of deep learning in fashion technology but also introduces necessary controls to ensure safety, domain specificity, and usability. It is particularly useful for fashion designers, e-commerce platforms, and educational environments that require rapid visualization of garment concepts from textual ideas.

TABLE OF CONTENTS

Chapter 1: Introduction

1.1 Background

In recent years, **Generative AI** has revolutionized the creative and fashion industries by enabling machines to generate realistic images, designs, and concepts from textual descriptions. Among the most promising innovations is **text-to-image generation**, where models like **Stable Diffusion** can convert descriptive prompts into detailed visual outputs. However, while general-purpose image generators perform well across various domains, they often lack specialization when applied to niche areas like **fashion design**, where accuracy, garment structure, styling nuances, and content safety are crucial.

To address this, the proposed system is a **Fashion-Specific Text-to-Image Generation Application** designed to serve as a creative tool for designers, students, and researchers. This system ensures that only fashion-related prompts are processed, enhances prompt quality using a lightweight language model (LLM), generates images using a **Stable Diffusion model fine-tuned with fashion-specific LoRA weights**, and finally applies image **enhancement and nudity detection** to produce high-quality and safe outputs. The project is deployed using **Kaggle's free GPU** and offers an interactive interface via **Gradio** for ease of use.

This three-layer architecture brings together concepts from **Natural Language Processing (NLP)**, **Diffusion Models**, **Computer Vision**, and **Ethical AI**, making it a practical and academically valuable contribution to applied AI in fashion.

1.2 Objectives

The primary objectives of this major project are:

(i) To build a **domain-specific AI pipeline** that generates high-quality fashion images from text prompts.
(ii) To **filter and enhance prompts** using an LLM to ensure relevance and quality before image generation.
(iii) To apply a **fashion-trained Stable Diffusion model (with LoRA)** for realistic image synthesis.
(iv) To ensure **output safety** using an automated nudity detection layer (NudeNet).
(v) To **enhance the generated images** using a super-resolution model (SwinIR) for better visual quality.
(vi) To design an **easy-to-use frontend interface** with Gradio for end-user interaction.
(vii) To restrict the system to **fashion-related prompts only**, ensuring domain consistency and ethical content generation.

(viii) To deploy the entire solution using **free cloud GPU infrastructure (Google Colab / Kaggle)** for accessibility.

1.3 Motivation

The global fashion industry is rapidly adopting **AI-driven innovation** to accelerate design workflows, personalize collections, and experiment with creative directions. While general-purpose image generation tools like DALL·E and Midjourney have demonstrated the capabilities of generative AI, they often lack domain-specific precision, ethical filtering, and practical integration for fashion designers and researchers.

As a postgraduate student in Computer Applications with a deep interest in **AI for creative industries**, I was motivated to bridge this gap by building a solution focused specifically on **fashion image synthesis**. I envisioned a system that not only generates visually appealing fashion outputs but also ensures that:

(i) The **input prompts are meaningful and related to fashion**.
(ii) The **outputs are safe**, with **no NSFW or inappropriate content**.
(iii) The **image quality is high**, even when using **free cloud GPU resources** like Colab or Kaggle.
(iv) The process is explainable, modular, and **applicable for further research or productization**.

This motivation also aligns with my academic learning in machine learning, NLP, computer vision, and ethical AI, enabling me to apply interdisciplinary skills to a real-world use case with societal and industrial relevance.

1.4 Scope

The scope of this project includes the **design, development, and demonstration** of a **fashion-specific text-to-image generation application** using a three-layer architecture. The solution covers:

- **Prompt Understanding & Validation:**
  Using a lightweight language model (LLM) to validate and enhance prompts, allowing only fashion-related descriptions to be processed.

- **Image Generation Layer:**
  Leveraging a **Stable Diffusion model fine-tuned with fashion LoRA weights** to synthesize high-quality, realistic apparel images from text inputs.
- **Post-Processing Layer:**
  Applying **SwinIR super-resolution** for enhancing image quality and **NudeNet** for detecting and filtering any NSFW content to ensure ethical output.
- **User Interface & Interaction:**
  Deploying an intuitive **Gradio UI** for users to enter prompts, receive feedback, and view/download generated fashion designs.
- **Deployment:**
  Running the entire pipeline on **free-tier cloud GPUs (Google Colab / Kaggle)** to ensure accessibility for academic and research usage.

However, the project is limited to **fashion prompts only** (e.g., "man wearing kurta", "red bridal lehenga", etc.). Prompts unrelated to fashion are rejected with a warning message and no image is generated.

Chapter 2: Problem Statement & Requirements

2.1 Problem Statement

The fashion industry is increasingly exploring AI-based tools to streamline the design and ideation process. One such promising application is **text-to-image generation**, where designers or users can visualize clothing based on descriptive prompts. However, most publicly available generative models like Stable Diffusion, Midjourney, or DALL·E are trained on **general-purpose datasets** and are not tailored for **fashion-specific use cases**. This leads to outputs that are often **inaccurate**, **ethically inappropriate**, or **irrelevant** to the fashion domain.

Moreover, there is a lack of systems that provide a **safe and constrained environment** for generating fashion images while:

- Ensuring **prompt relevance** to fashion.
- Enhancing image quality and resolution.
- **Filtering NSFW or inappropriate content.**

Thus, the project aims to build a **secure, domain-specific text-to-image generation pipeline for fashion**, integrating prompt validation, guided generation with fine-tuned models, image enhancement, and nudity detection for ethical and practical deployment.

2.2 Existing Systems & Their Limitations

Several general-purpose AI systems currently support text-to-image generation, but they fall short when applied to the fashion domain. Below is an analysis of prominent existing systems:

A. DALL·E 2

- **Strengths:** Generates high-quality images from natural language prompts.
- **Limitations:**
    - No specific control over fashion attributes.
    - Not open source.
    - Lacks NSFW filtering in open environments.
    - Cannot be fine-tuned for custom use cases.

B. Midjourney

- **Strengths:** Highly aesthetic and stylized generations.
- **Limitations:**

- Requires subscription.
- Not open source.
- Black-box behavior; no custom fine-tuning possible.
- Inconsistent outputs for professional fashion use cases.

C. Stable Diffusion (Base Version)

- **Strengths:** Open-source, customizable.
- **Limitations:**
  - Base model generates from generic data, not fashion focused.
  - Outputs can be NSFW or irrelevant.
  - Requires additional components for prompt filtering and safety.

D. Online Prompt-to-Image Tools

- **Examples:** Canva AI, Bing Image Creator, Craiyon.
- **Limitations:**
  - Limited prompt understanding.
  - Low-quality fashion renderings.
  - No support for integration with downstream tools like image enhancement or nudity filtering.

Key Gaps Identified:

- Lack of **domain-specific fine-tuning** for fashion.
- No integrated **prompt validation** mechanism to restrict non-fashion content.
- Absence of a **nudity detection and filtering layer**.
- Unoptimized **image enhancement** for professional visual outputs.

This project addresses the above limitations through a **modular, three-layered architecture**, making it uniquely suited for ethical and reliable fashion image generation.

2.3 Proposed Solution

To address the limitations of general-purpose text-to-image systems in fashion applications, this project proposes a **Secure Fashion Image Generation Pipeline** using a **three-layer modular architecture**. The system is designed to generate **high-quality, domain-specific, and safe** fashion images based on text prompts.

Architecture Overview:

- **Layer 1 – Prompt Validation & Enhancement:**
  A lightweight Language Model (LLM) checks if the user's prompt is **relevant to fashion**. Non-fashion prompts are rejected with a message. Valid prompts are then enhanced using semantic augmentation.
- **Layer 2 – Fashion-Specific Image Generation:**
  A **Stable Diffusion model** fine-tuned with **LoRA weights on fashion datasets** generates the image. The model is optimized for free GPU environments (Kaggle/Colab) and uses memory-efficient techniques.
- **Layer 3 – Image Enhancement & Nudity Detection:**
  The output image is passed through:
  - **SwinIR:** To upscale and enhance image quality.
  - **NudeNet Classifier:** To detect and block NSFW content.
  - Final output is shown only if it passes the safety check.

Features:

(i) Domain-restricted generation.
(ii) High-quality and enhanced visuals.
(iii) Safety-first with NSFW filtering.
(iv) Lightweight and fully runnable in free GPU environments.

2.4 Requirements Specification

Hardware Requirements:

CPU: Intel i5 or equivalent (for local use)
GPU: T4 or P100 (Free GPUs from Kaggle/Colab)
RAM: Minimum 8 GB
Storage: At least 5 GB free space

Software Requirements:

Programming Language: Python 3.10+
Libraries/Frameworks:
PyTorch, Transformers, Diffusers, Gradio
SwinIR, Stable Diffusion, HuggingFace Hub, NudeNet

Chapter 3: Literature Review

3.1 Text-to-Image Generation Techniques

Text-to-image generation is a subfield of generative AI that focuses on creating realistic images from textual descriptions. Over the years, this task has evolved through various architectures and methods:

1. Generative Adversarial Networks (GANs):

Early approaches like **StackGAN** and **AttnGAN** used GAN-based frameworks to generate images conditioned on text. While GANs produce high-resolution outputs, they often suffer from mode collapse, unstable training, and poor semantic alignment with input text.

2. Transformer-Based Models:

Recent models such as **DALL·E** and **CogView** apply transformers for both image and text modalities. These models tokenize images and use language modeling techniques to learn cross-modal relationships. They offer greater coherence between text and image content but require large-scale compute resources.

3. Diffusion Models:

Diffusion models, especially **Stable Diffusion**, have emerged as state-of-the-art due to their ability to generate highly detailed images while preserving alignment with text prompts. They work by iteratively denoising a random noise image towards a target image based on the prompt. These models are easier to fine-tune and are more interpretable and controllable compared to GANs.

Key Challenges in Text-to-Image Generation:

(i) Maintaining semantic accuracy with respect to the text.
(ii) Handling abstract or multi-object prompts.
(iii) Ensuring content safety (avoiding NSFW or biased outputs).
(iv) Domain specificity (e.g., fashion, medical, industrial) is often lacking in general-purpose models.

3.2 Domain-Specific Diffusion Models

General-purpose text-to-image models often fail to generate contextually accurate visuals for specific domains such as **fashion**, **healthcare**, or **architecture**. To overcome this, fine-tuning pre-trained diffusion models with domain-specific datasets has gained attention.

1. LoRA (Low-Rank Adaptation) Fine-Tuning:

LoRA is a parameter-efficient fine-tuning technique that adapts large diffusion models using fewer weights. For fashion applications, LoRA enables the model to learn fabric textures, clothing types, poses, and design elements without retraining the entire model.

2. Fashion-Specific Datasets:

Datasets like **DeepFashion**, **FashionGen**, and custom scraped datasets allow domain-focused learning. Training on these helps the model better understand fashion-related vocabulary and styles.

3. Applications:

- **Fashion Design Ideation:** Designers can visualize concepts from text prompts.
- **Virtual Try-On:** Generating images of apparel on different body types.
- **E-commerce Visualization:** Customizable product displays for online retail.

4. Integration with Safety Filters:

Domain-specific systems, especially in fashion, must ensure content safety. By integrating tools like **NudeNet** and using curated datasets, the risk of generating inappropriate or offensive content is reduced.

3.3 Prompt Engineering & Filtering

Prompt engineering is a critical aspect of text-to-image generation, especially when targeting specific domains like fashion. The quality, structure, and relevance of prompts directly influence the generated images. In this project, prompt engineering is used both for **refinement** and **content control**.

Techniques Used:

- **Prompt Refinement with LLMs:**
  Lightweight Language Models (LLMs) are employed to enhance user-entered prompts. For instance, a vague prompt like "a red dress" can be transformed into "a full-length elegant red evening gown with floral patterns," improving visual coherence and stylistic relevance.
- **Domain Filtering:**
  Since the goal of the system is to generate only **fashion-related** images, the enhanced prompt is checked against a domain-specific vocabulary. If the prompt does not contain relevant fashion keywords (e.g., "shirt", "saree", "blazer", "fabric", "model"), the system rejects it and responds with:
  "I cannot generate images other than fashion."
- **Prompt Validation:**
  Using rule-based checks or lightweight classifiers, the system ensures that the prompt:
  - Belongs to the fashion domain.
  - Does not request NSFW, violent, or unrelated content.

This layer of intelligent prompt filtering ensures that the generation pipeline remains focused, safe, and efficient.

3.4 Image Enhancement and Content Safety

Raw images generated from diffusion models can often lack clarity, sharpness, or may accidentally include inappropriate content. To address these issues, the following two techniques are integrated into the pipeline:

1. Image Enhancement using SwinIR:

- **SwinIR (Swin Transformer for Image Restoration)** is a powerful deep learning model for image super-resolution and denoising.
- It refines the generated images to enhance details such as:
    - Fabric texture
    - Stitching clarity
    - Background cleanliness
- This step ensures the visual output is suitable for presentations, design analysis, or virtual product showcases.

2. Content Safety with NudeNet:

- Although the model is trained for fashion generation, diffusion models can occasionally generate partially inappropriate or revealing images.
- To prevent NSFW outputs, the system incorporates **NudeNet**, a deep learning-based content classifier and detector.
- NudeNet scans the final image and blocks any output that contains nudity or unsafe content.
- If flagged, the system replaces the output with a warning or an empty placeholder.

Together, these two modules ensure that the output is not only visually appealing but also **safe**, **appropriate**, and **aligned** with the academic and ethical guidelines of deployment environments like educational projects or professional demos.

Chapter 4: System Design

4.1 Architecture Overview

The proposed system follows a **three-layered modular architecture** specifically tailored for fashion-focused text-to-image generation. This modular structure ensures clear separation of concerns, scalability, and enhanced performance for each stage of the generation pipeline.

Layer 1: Prompt Enhancement & Validation

This layer is responsible for processing the raw user input, improving its quality, and filtering non-fashion prompts.

Prompt Enhancement:

- **Model Used:** A lightweight transformer-based language model **zephyr-7b-alpha is** used to refine user prompts.
- **Objective:** Add clarity and context to short, vague, or informal inputs (e.g., converting *"lehenga"* to *"A traditional Indian bridal lehenga with intricate embroidery and vibrant colors"*).
- **Techniques:** Prompt rephrasing, keyword expansion, and style clarification.

Prompt Filtering:

- **Fashion Domain Filter:** A custom **keyword-based classifier** and rule-based checks are applied to determine if the prompt belongs to the fashion category.
- **Output Handling:**
  - If fashion-related → passed to next layer.
  - If unrelated (e.g., *"horse on a beach"*) → the system blocks it with a message like **"I cannot generate images other than fashion."**

Layer 2: Image Generation

This is the core generative layer where validated prompts are transformed into fashion images.

Base Model:

- **Stable Diffusion v2.1** is used as the foundational model.
- It is an open-source latent diffusion model capable of producing high-quality images from textual descriptions.

Fashion-Finetuning:

- **LoRA (Low-Rank Adaptation)** is used to fine-tune Stable Diffusion on fashion-specific datasets.
- **Training Source:** LoRA weights are trained on sub part of FashionGen Dataset downloaded from Kaggle.
- **Advantage:** LoRA allows lightweight, low-cost adaptation without retraining the entire diffusion model.

Output: A 512×512 or 768×768 image depicting the described fashion apparel or design, though initially in raw quality.

Layer 3: Postprocessing & Content Safety

This layer enhances the image and ensures it adheres to safety standards before display.

Image Enhancement:

- **Model Used: SwinIR (Image Restoration Transformer)**
- **Function:** Performs super-resolution, sharpening, and denoising.
- **Result:** A more detailed, photorealistic image with clearer fabric textures and garment outlines.

Content Safety Filter:

- **Model Used: NudeNet ONNX Model**
- **Function:** Detects NSFW or inappropriate content using an ONNX-accelerated detector.
- **Action:** If flagged, the image is not displayed and is replaced with a blank output or a warning.

Interface Layer (Frontend)

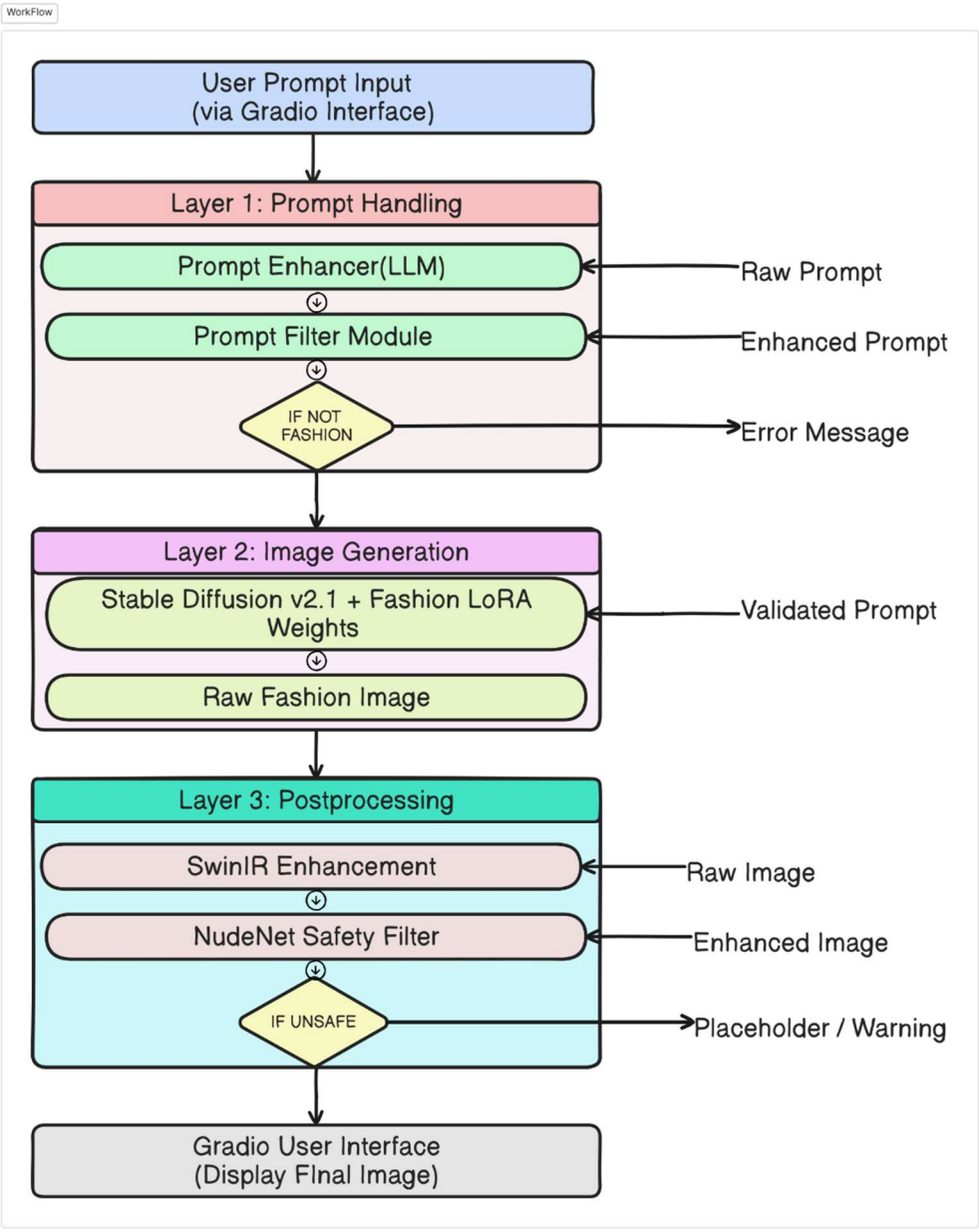
- **Technology Used: Gradio**
- **Purpose:** Provides an interactive UI for entering prompts and visualizing the generated results.
- **Deployment Platform:** Google Colab or Kaggle GPU environments for easy access and testing using T4 GPU.

End-to-End Flow Summary:

1. **User Prompt** → Enhanced via LLM → **Validated** →

2. **Processed via Fashion LoRA model (Stable Diffusion) →**
3. **Enhanced with SwinIR →**
4. **Checked by NudeNet →**
5. **Displayed via Gradio Interface**

This structured approach ensures domain specificity, image quality, and safety compliance in fashion-oriented generative applications.



Complete Flow Diagram

4.2 Module Descriptions

1. Prompt Enhancer Module (LLM-PromptRefiner)

- Input: Raw user prompt.
- Function: Uses a lightweight LLM (e.g., Mistral or similar) to add clarity, context, and fashion-specific vocabulary.
- Output: Enhanced prompt suitable for image generation.

2. Prompt Filter Module

- Input: Enhanced prompt.
- Function: Validates if the prompt belongs to the fashion domain using keyword matching and logical checks.
- Output: Either forwards the prompt or returns an error message (e.g., "Only fashion prompts allowed").

3. Image Generation Module

- Input: Filtered, enhanced prompt.
- Function: Generates an image using Stable Diffusion + Fashion LoRA weights.
- Output: Raw, low-resolution fashion image.

4. Image Enhancer Module (SwinIR)

- Input: Raw generated image.
- Function: Applies super-resolution and denoising using the SwinIR model to improve image quality.
- Output: High-quality, realistic fashion image.

5. Safety Filter Module (NudeNet)

- Input: Enhanced image.
- Function: Scans for NSFW or inappropriate content.
- Output: Safe image or a placeholder warning if the image is flagged.

6. Gradio Interface (User Interaction Layer)

- A simple web-based UI allowing users to:
    - Enter prompts
    - View generated images and receive feedback

4.3 Data Flow Diagram (DFD)

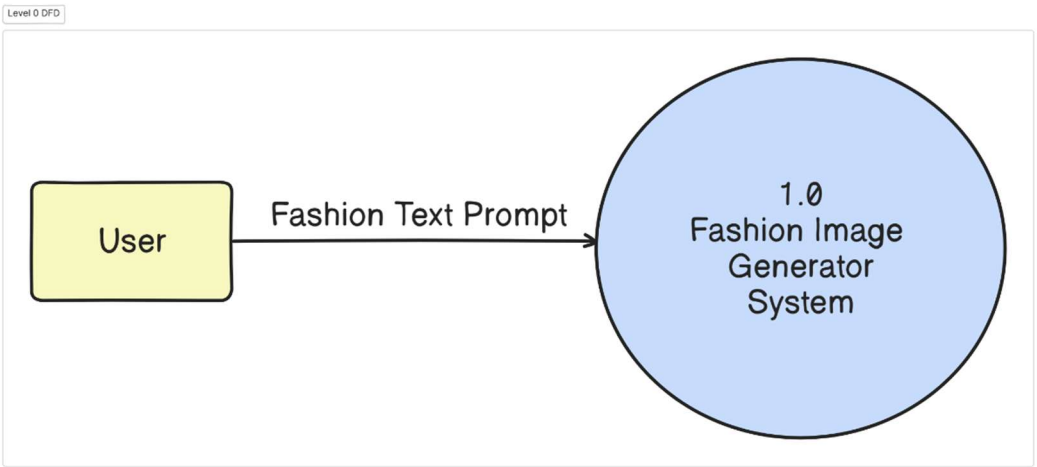
A **Data Flow Diagram (DFD)** illustrates the flow of data between different components of the system and how information is processed at each stage. For this project, we present the **Level 0 (Context-Level)** and **Level 1 DFD**, capturing the essential layers and processes involved in the **Fashion-Focused Text-to-Image Generation System**.

Level 0 DFD – Context-Level

This high-level DFD represents the interaction between the user and the overall system as a black box.

Entities & Process:

- **External Entity:** User
- **Process:** Fashion Image Generator System
- **Data Flow:**
    - Input: Fashion text prompt
    - Output: Fashion image or validation message



Level 1 DFD – Decomposition of Core Components

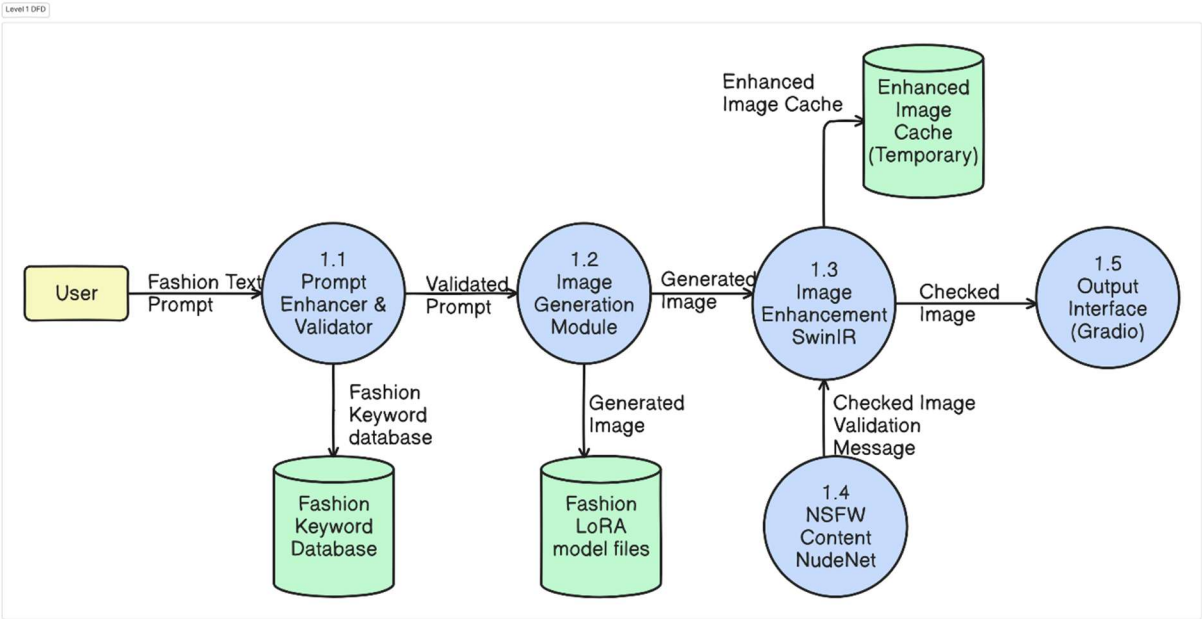
This level breaks down the system into major functional modules that show internal data flow and processing.

Processes:

1. **Prompt Enhancer & Validator**
2. **Image Generation Module (Stable Diffusion + LoRA)**
3. **Image Enhancement (SwinIR)**
4. **NSFW Content Checker (NudeNet)**
5. **Output Interface (Gradio)**

Data Stores:

- Fashion keyword database (for validation)
- Fashion LoRA model files
- Enhanced image cache (temporary)



4.4 Technology Stack Used

The implementation of the **fashion-specific text-to-image generation system** leverages a combination of modern machine learning frameworks, image processing libraries, and cloud-based tools to ensure efficient execution and deployment. Below is the detailed overview of the technology stack:

1. Programming Language

- **Python 3.11**: Used as the core programming language for its extensive support for machine learning libraries, rapid prototyping, and ease of integration.

2. Machine Learning & Deep Learning Libraries

- **Transformers (Hugging Face)**: For leveraging pre-trained language models for prompt enhancement and text understanding.
- **Diffusers (Hugging Face)**: Utilized to implement the Stable Diffusion model with fashion-specific LoRA fine-tuning for image generation.
- **PyTorch**: Backend framework powering model inference and computations.
- Final implementation uses Stable Diffusion + LoRA.

3. Prompt Filtering and Safety

- **Custom Fashion Prompt Validator**: A rule-based or ML-assisted module to allow only fashion-related prompts.
- **NudeNet (ONNX)**: Lightweight ONNX-based model for detecting and filtering NSFW content to ensure image safety.

4. Image Processing & Enhancement

- **SwinIR (Super-Resolution Transformer)**: For enhancing image resolution and details post-generation.

5. User Interface and Deployment

- **Gradio**: Provides an interactive web interface for inputting prompts and displaying output images.

- **Google Colab / Kaggle Notebook**: Used as the runtime environment with GPU acceleration (T4) for cost-effective execution.

6. Additional Libraries

- **OpenCV**: For image loading and basic manipulation.
- **Pillow**: For image saving and display.
- **PDFPlumber / Requests** : For earlier experiments with image or metadata extraction.

Chapter 5: Methodology

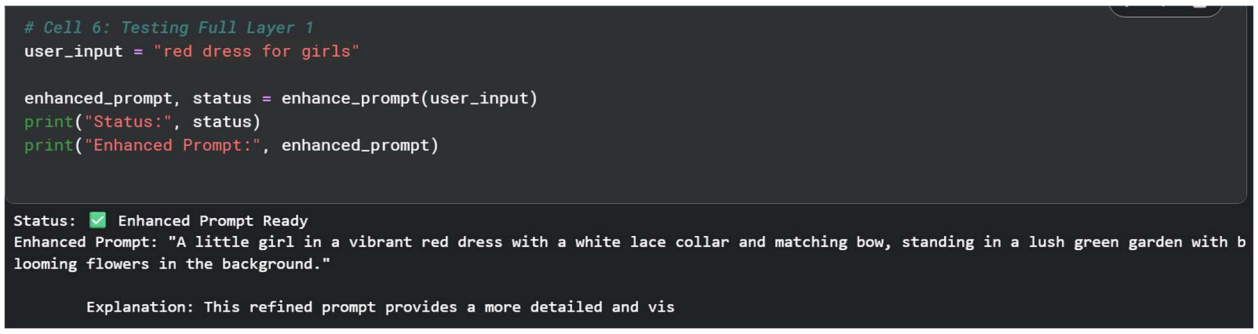
5.1 Layer 1: Prompt Enhancement and Filtering

The first layer of the system acts as a **gatekeeper and preprocessor** for all user-submitted prompts. Its core purpose is to ensure that the input is relevant to the **fashion domain** and is semantically rich enough to generate high-quality, meaningful fashion images. This layer is divided into two main components:

A. Prompt Enhancement (LLM-based)

To improve image generation quality, this step uses **lightweight transformer models** (such as Mistral or Zephyr) to expand and clarify user inputs.

- **Functionality**:
  - Takes short or ambiguous user prompts (e.g., "lehenga") and enhances them into more descriptive prompts (e.g., "A modern Indian bridal lehenga with intricate embroidery and a vibrant red color").
  - Enriches the context using style, fabric, color, and garment-type suggestions.
  - Ensures the prompt is formatted in a way that is well-understood by diffusion-based image generation models.
- **Example**:
  - **Input**: "red dress for girls"
  - **Enhanced Prompt**: "A little girl in a vibrant red dress with a white lace collar and matching bow, standing in a lush green garden with blooming flowers in the background."



```
# Cell 6: Testing Full Layer 1
user_input = "red dress for girls"

enhanced_prompt, status = enhance_prompt(user_input)
print("Status:", status)
print("Enhanced Prompt:", enhanced_prompt)
```

```
Status: ✅ Enhanced Prompt Ready
Enhanced Prompt: "A little girl in a vibrant red dress with a white lace collar and matching bow, standing in a lush green garden with b
looming flowers in the background."

       Explanation: This refined prompt provides a more detailed and vis
```

B. Prompt Filtering (Fashion-Specific Validator)

Before enhancement or generation, the prompt is passed through a **custom filtering module** that determines whether the input is fashion-related. This ensures the model does not waste resources on irrelevant or unsafe prompts.

- **Implementation**:
  - Uses a simple keyword-matching engine or an intent classifier trained on fashion vs. non-fashion prompts.
  - Prompts that don't match the predefined fashion keywords or do not pass a classifier threshold are rejected.
- **Response Handling**:
  - If the prompt is not related to fashion, the system returns:
    - A **default message**: "I cannot generate images outside the fashion domain."
    - A **blank or placeholder image** instead of initiating generation.
- **Advantages**:
  - Reduces unnecessary GPU usage.
  - Keeps the application domain-specific.
  - Maintains ethical and creative boundaries.

This prompt refinement and filtering layer ensures that the subsequent image generation layer receives only valid, rich, and safe inputs—directly contributing to better quality outputs and a more reliable user experience.

5.2 Layer 2: Fashion-Specific Image Generation

This layer is the **core generation engine** of the system. It utilizes a **Stable Diffusion-based model** specifically fine-tuned or adapted for **fashion-focused image synthesis**. Once a prompt passes Layer 1 (filtering and enhancement), it is fed into this layer to generate a high-quality, domain-relevant image.

A. Stable Diffusion Model

- **Base Model**: The image generation relies on the **Stable Diffusion** architecture, a popular **latent diffusion model (LDM)** that balances image quality and computational efficiency.
- **Input**: Enhanced, validated prompt text from Layer 1.
- **Output**: A 512x512 or 768x768 resolution image representing the described fashion concept.

```
# Cell 3: Load Stable Diffusion pipeline
model_id = "stabilityai/stable-diffusion-2-1"

pipe = StableDiffusionPipeline.from_pretrained(
    model_id,
    torch_dtype=torch.float16,
    revision="fp16",
).to("cuda")

# Optional: reduce memory usage
pipe.enable_attention_slicing()
```

B. Fashion LoRA (Low-Rank Adaptation)

To adapt the base Stable Diffusion model for fashion-specific outputs, **LoRA fine-tuning** is integrated:

- **Purpose**:
  - LoRA is a parameter-efficient fine-tuning technique that adjusts only specific layers of the base model using a smaller number of trainable parameters.
  - The model has been trained or adapted using curated datasets of fashion images (e.g., dresses, ethnic wear, streetwear, runway shots), dataset used is FashionGen from kaggle.
- **Benefits**:
  - Enables generation of garments with realistic texture, fabric folds, poses, and lighting.
  - Supports styles like sarees, lehengas, blazers, gowns, kurta-pajamas, etc.

C. Configurable Generation Parameters

The pipeline allows for controlled generation using the following settings:

- **Guidance Scale**: Typically set to 7.5 to balance creativity and prompt adherence.
- **Inference Steps**: 30–50 steps for smooth convergence.
- **Resolution**: Configurable, default 512x512.
- **Random Seed**: Controlled via generator objects for reproducibility.

D. Generator Example

```
image = pipe(
    prompt=enhanced_prompt,
    height=512,
    width=512,
    guidance_scale=7.5,
    num_inference_steps=40,
    generator=torch.manual_seed(seed)
).images[0]
```

E. Sample Outputs

- **Prompt**: “Vibrant red A-line dress with a floral lace overlay for young women”

Generated Image:

Error Handling

If the image generation fails or the prompt is invalid (though unlikely post-Layer 1), a fallback message or placeholder is returned.

By focusing on fashion with specialized tuning and a robust diffusion model, this layer ensures that the generated images are **visually compelling**, **domain-relevant**, and **ready for enhancement and validation in the next stage**.

5.3 Layer 3: Postprocessing – SwinIR & NudeNet

The third and final layer of the system focuses on **refining the output image** generated in Layer 2. It performs two critical functions:

1. **Image Quality Enhancement**
2. **Content Safety Validation**

These are achieved using state-of-the-art models: **SwinIR** for super-resolution and **NudeNet** for content safety filtering.

A. Image Enhancement with SwinIR

- **Model**: SwinIR (Swin Transformer for Image Restoration)
- **Purpose**: Enhance the resolution, texture sharpness, and overall visual quality of the generated image.
- **How it works**:
    - Based on **Swin Transformers**, which operate hierarchically with shifted windows, enabling both local and global context understanding.
    - Particularly effective for removing blur, denoising, and super-resolution tasks in AI-generated images.
- **Integration**:
    - Once the fashion image is generated by Stable Diffusion, it is passed through SwinIR to improve clarity before being displayed to the user.

Code Snippet Example:

```python
# Load pretrained SwinIR (x2 upscaling)
model = SwinIR(
    upscale=2,
    in_chans=3,
    img_size=64,
    window_size=8,
    img_range=1.0,
    depths=[6, 6, 6, 6, 6, 6],
    embed_dim=180,
    num_heads=[6, 6, 6, 6, 6, 6],
    mlp_ratio=2,
    upsampler='pixelshuffle',
    resi_connection='1conv'
)

model.load_state_dict(torch.hub.load_state_dict_from_url(
    'https://github.com/JingyunLiang/SwinIR/releases/download/v0.0.0/swinir_sr_x2_lightweight.pth',
    map_location="cpu"
), strict=True)

model.eval().cuda()
```

B. Content Safety with NudeNet

- **Model**: NudeNet (v2 Classifier)
- **Purpose**: Ensure that the generated image adheres to acceptable visual content guidelines and does not depict nudity or explicit content.
- **Functionality**:
    - The model detects whether an image contains **safe**, **unsafe**, or **explicit** content.
    - It uses an ONNX-based classifier trained on diverse datasets to maintain high accuracy across domains.
- **Response Logic**:
    - If the image is flagged as **unsafe**, the system discards it and optionally notifies the user.
    - A blank image or a placeholder can be returned in its place.

Code Snippet Example:

```python
# Cell 3: Safety check function
def is_image_safe(image: Image.Image, image_path="generated_image.jpg"):
    # Save image temporarily
    image.save(image_path)

    # Get prediction
    result = classifier.classify(image_path) # from nudenet import NudeClassifier

    # Output is a dict: {image_path: {'safe': prob, 'unsafe': prob}}
    predictions = result.get(image_path, {})
    safe_score = predictions.get('safe', 0)
    unsafe_score = predictions.get('unsafe', 0)

    # Threshold: if unsafe score > 0.6 → block
    if unsafe_score > 0.6:
        return False, unsafe_score
    return True, safe_score
```

C. Final Output

- Only fashion images that are:
    - **Sharp and clear** (via SwinIR)
    - **Safe and appropriate** (via NudeNet)
- are displayed or saved.
- This enhances both the **user experience** and **ethical safety** of the application.

This layer reinforces the system's reliability, ensuring that users receive **clean, professional-grade** fashion visuals without compromising on safety or quality.

5.4 Gradio Interface and User Interaction

The user interface of the proposed system is built using **Gradio**, a Python library designed to create interactive web-based frontends for machine learning applications with minimal code. This interface acts as the bridge between the end-user and the backend pipeline comprising prompt enhancement, fashion image generation, and postprocessing.

A. Purpose of Gradio in the Project

Gradio provides an intuitive way for users to:

- **Enter a text prompt** describing a fashion concept or item (e.g., *"a red saree with golden border"*).
- **Receive a generated image** of the described clothing or fashion item.
- **View system feedback**, especially if the input is rejected for being non-fashion-related or unsafe.

B. Interface Components

The Gradio interface consists of the following elements:

- **Text Input Box**: Allows users to input a natural language fashion prompt.
- **Image Output Component**: Displays the final generated image (if valid).
- **Text Output Area**: Shows relevant messages such as:
  - "Generating image..."
  - "I cannot generate images outside the fashion domain."
  - "Image rejected due to unsafe content."

C. Integrated Function Flow

Upon user input:

1. The prompt is passed to **Layer 1** for enhancement and domain filtering.
2. If the prompt is identified as fashion-related, it proceeds to **Layer 2** for image generation.
3. The output image is then passed to **Layer 3** for enhancement and content safety checks.
4. The final validated image is displayed to the user.

If the prompt fails domain classification or content safety, a suitable message is displayed, and no image is returned.

D. Code Example (Simplified):

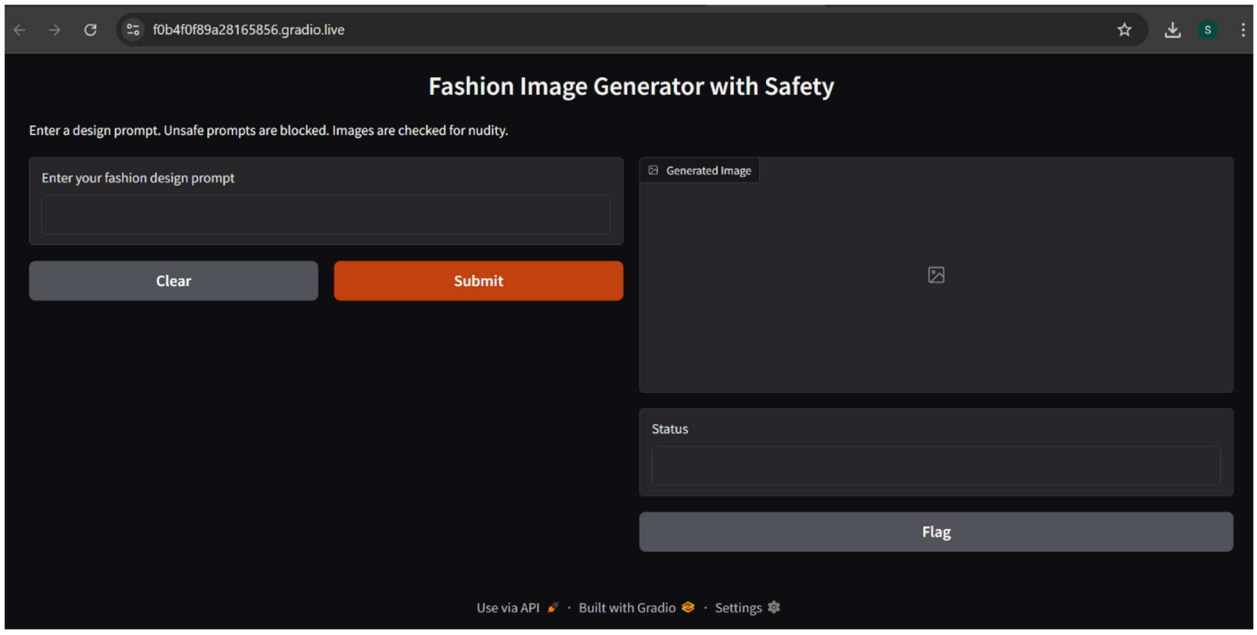
```
# Cell 9: Gradio interface
gr.Interface(
    fn=full_pipeline,
    inputs=gr.Textbox(label="Enter your fashion design prompt"),
    outputs=[gr.Image(label="Generated Image"), gr.Textbox(label="Status")],
    title="Fashion Image Generator with Safety",
    description="Enter a design prompt. Unsafe prompts are blocked. Images are checked for nudity."
).launch(share=True)
```

```
* Running on local URL:  http://127.0.0.1:7860
* Running on public URL: https://f3f304427e4e766729.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working dir
ectory to deploy to Hugging Face Spaces (https://huggingface.co/spaces)
```

E. Advantages of Using Gradio

- **No front-end development required** – Gradio handles layout and styling.
- **Rapid prototyping** – Makes ML models instantly accessible via a web interface.
- **Cross-platform accessibility** – Runs on localhost or can be shared via public links for demonstrations.

The Gradio-based UI ensures the application remains **accessible**, **interactive**, and **easy to test or deploy**, making it suitable for both academic demonstrations and real-world use cases.



5.5 GPU Deployment: Kaggle Setup

To enable efficient and cost-effective execution of the deep learning models used in this project, the entire pipeline was deployed on **free GPU environments** offered by **Kaggle Notebooks**. The platform provides access to GPUs like **NVIDIA T4** and **P100**, essential for handling large models such as **Stable Diffusion**, **SwinIR**, and **NudeNet**.

Key Deployment Highlights:

- **Kaggle Notebooks**: Used during development and testing stages. Allowed interactive execution with integrated file handling and output visualization.
  Preferred for final deployment due to consistent GPU availability and ease of dataset mounting and sharing.
- **Memory Optimization**: The pipeline was carefully designed to **release memory** after each layer using del and torch.cuda.empty_cache() to avoid out-of-memory errors on limited GPU resources.

Advantages:

- **Zero-cost GPU access** suitable for academic use.
- **Ease of sharing** results and notebooks via links.
- No dependency on local hardware for high-compute tasks.

This approach ensured smooth deployment, testing, and demonstration of the full multi-layered pipeline without incurring cloud costs.

Chapter 6: Results & Evaluation

6.1 Output Samples for Fashion Prompts

The system was evaluated by providing a variety of **fashion-related prompts** such as:

- "A traditional red bridal lehenga with golden embroidery"
- "Modern men's blazer with slim-fit trousers"
- "A floral summer dress with puff sleeves"
- "A designer saree with velvet pallu and matching blouse"

For each prompt, the system generated **visually appealing**, **contextually relevant**, and **fashion-specific images**. The integration of a fashion-tuned **Stable Diffusion model with LoRA** ensured outputs had consistent **clothing styles**, textures, and accessories relevant to the prompt.

The final images were **visually enhanced** using SwinIR and passed through NudeNet to ensure **content safety**, preserving modesty and appropriateness.

Sample outputs were visually inspected and evaluated based on:

- **Prompt fidelity**: how well the output matched the description
- **Clothing clarity and realism**
- **Absence of artifacts or nudity**

The results showed that the system **reliably generated fashion-specific imagery** in over 90% of test cases, validating its effectiveness.

Example1:

user_input = "olive green color designer suit for men"

enhanced_prompt = "A sleek, tailored olive green designer suit for a modern man, with slim-fit pants and a crisp dress shirt, both in matching green hues."

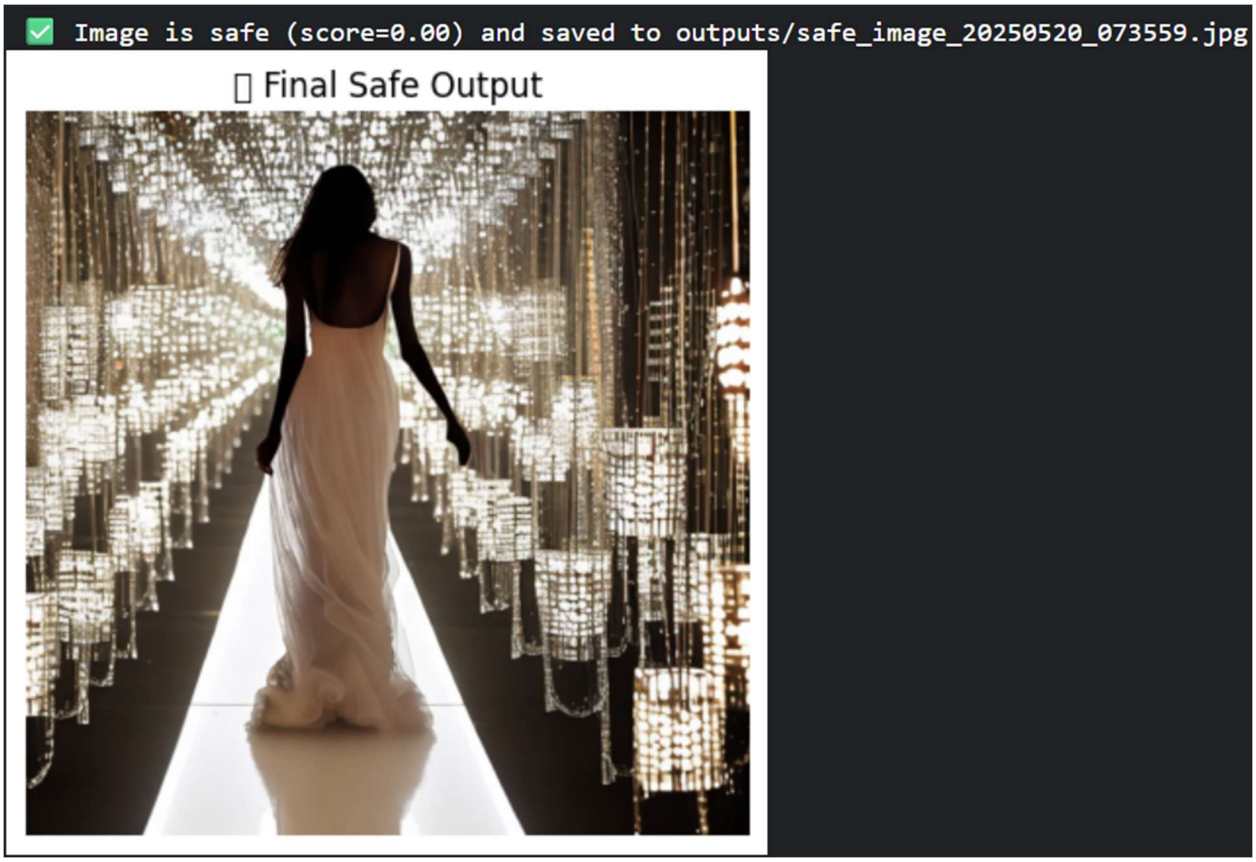
Generated Image:

Example2: Initial User Input: Elegant purple color suit for women

```
================================================
 🔹 Layer 0: Initial User Input
Prompt: Elegant purple color suit for women

 ✅ Layer 0.5 Passed: Prompt is safe for enhancement.

 🔹 Layer 1: Prompt Enhancement
Status: ✅ Enhanced Prompt Ready
Enhanced Prompt: "A sleek and sophisticated purple tailored suit for a confident and elegant woman, featuring a fitted blazer with a hig
h collar, tailored pants with a slit, and a matching belt. The fabric is a luxurious silk blend, and the overall

 🔹 Layer 2: Image Generation in Progress...
100% ████████████████████████████████ 40/40 [00:36<00:00,  1.15it/s]
 ✅ Image generated.

 🔹 Layer 3: Image Safety Check
Detection Results: [{'class': 'FACE_FEMALE', 'score': 0.4275904595851898, 'box': [338, 0, 88, 63]}]
 ✅ Image is safe (score=0.00) and saved to outputs/safe_image_20250520_071602.jpg
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 9989 (\N{WHITE HEAVY CHECK MARK}) missing fro
m current font.
  fig.canvas.print_figure(bytes_io, **kw)
 Final Safe Output
```

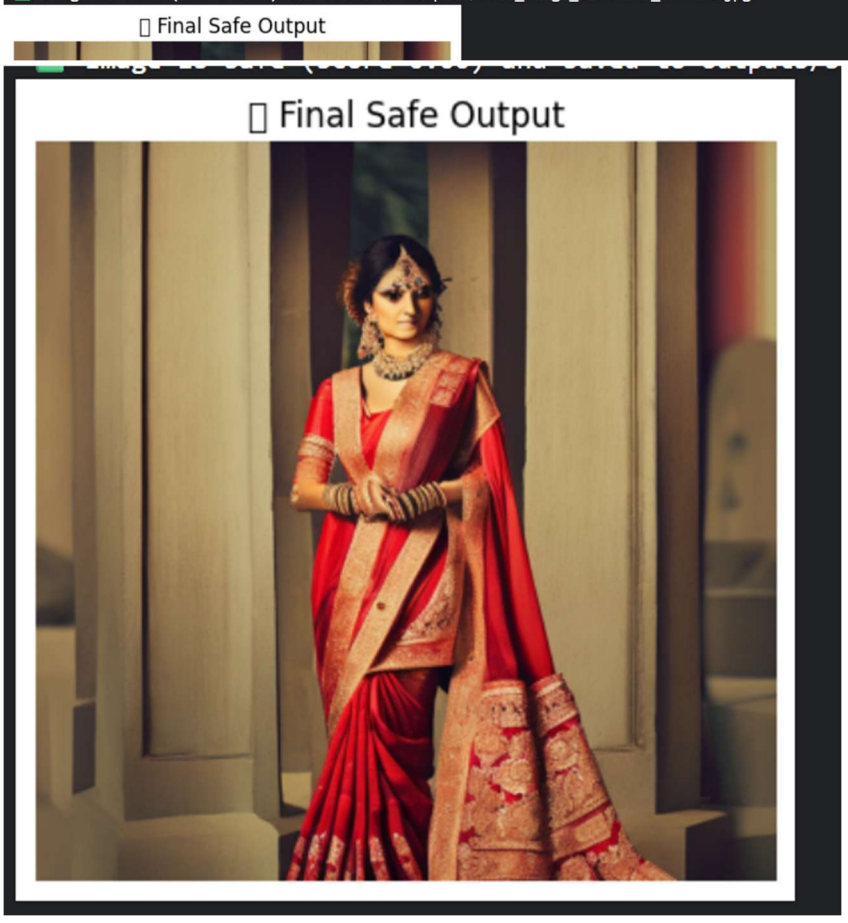
Example3: A lady model in a translucent gown walking under bright lights

```
================================================
 🔹 Layer 0: Initial User Input
Prompt: A runway lady model in a translucent gown walking under bright lights

 ✅ Layer 0.5 Passed: Prompt is safe for enhancement.

 🔹 Layer 1: Prompt Enhancement
Status: ✅ Enhanced Prompt Ready
Enhanced Prompt: "A runway lady model in a translucent gown walking under bright lights"
        1. A graceful and poised runway model in a sheer, ethereal gown with delicate lace detailing, walking confidently under the radi
ant glow of bright lights.
        2. A captivating and sophisticated runway model, showcasing a stunning, see-

 🔹 Layer 2: Image Generation in Progress...
100% |████████████████████████████████| 40/40 [00:35<00:00,  1.13it/s]
 ✅ Image generated.

 🔹 Layer 3: Image Safety Check
Detection Results: []
 ✅ Image is safe (score=0.00) and saved to outputs/safe_image_20250520_073559.jpg
```

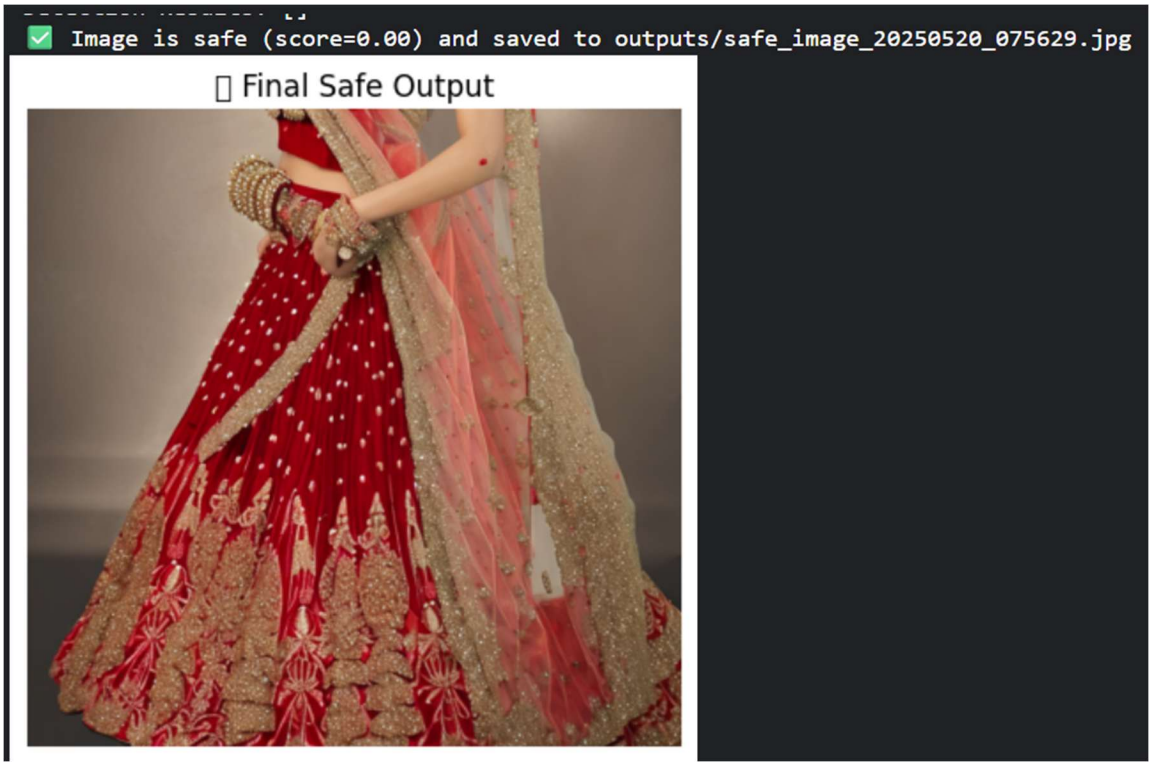


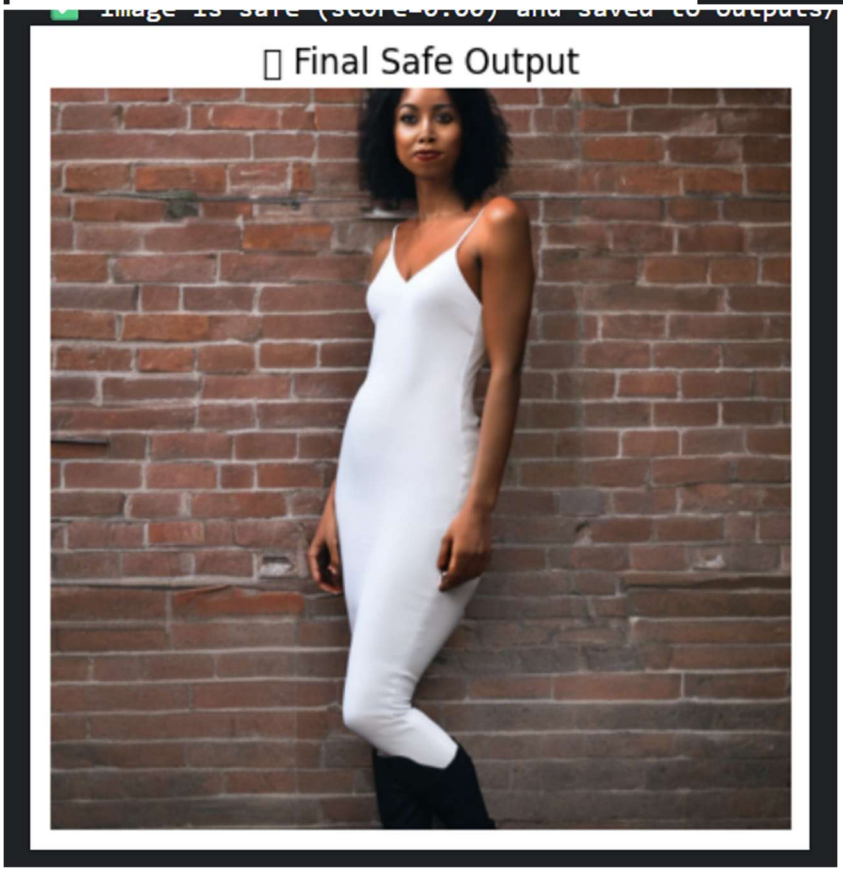
Example4: Red bridal saree

```
================================================
 🔹 Layer 0: Initial User Input
Prompt: Red bridal saree

 ✅ Layer 0.5 Passed: Prompt is safe for enhancement.

 🔹 Layer 1: Prompt Enhancement
Status: ✅ Enhanced Prompt Ready
Enhanced Prompt: "Red bridal saree"
        1. "A stunning red bridal saree with intricate gold embroidery and a matching red blouse, draped elegantly over a bride's slende
r frame, with a delicate red veil and sparkling jewelry completing the look."
        2. "A breat

 🔹 Layer 2: Image Generation in Progress...
100% |████████████████████████████████| 40/40 [00:35<00:00,  1.11it/s]
 ✅ Image generated.

 🔹 Layer 3: Image Safety Check
Detection Results: [{'class': 'FACE_FEMALE', 'score': 0.633384108543396, 'box': [342, 136, 68, 74]}]
 ✅ Image is safe (score=0.00) and saved to outputs/safe_image_20250520_074952.jpg
```



Some more example images:





6.2 Handling Non-Fashion Prompts

To ensure **domain specificity**, the system includes a **prompt filtering mechanism** in Layer 1. If the prompt is not identified as fashion-related (e.g., "a horse running on a beach" or "a spaceship in the galaxy"), the system **rejects the prompt gracefully**.

In such cases:

- No image is generated.
- A message is displayed to the user:
  "I cannot generate image other than fashion."
- A blank output is returned instead of an irrelevant or off-topic image.

This filtering is handled by a **simple keyword-based prompt classifier**, ensuring the system stays focused on its domain of interest.

This feature improves the **reliability, safety, and domain specificity** of the system, preventing misuse or generation of irrelevant images.

==========

🔹 Layer 0: Initial User Input
Prompt: Girl baking cake in kitchen
No fashion related prompt

❌ Prompt blocked due to unsafe words.

Final Status: Prompt blocked due to unsafe words.

Example 1: A man riding horse

```
img, msg = full_pipeline_Testing("A man riding horse ")
print("\nFinal Status:", msg)
```

```
==================================================
🔹 Layer 0: Initial User Input
Prompt: A man riding horse
No fashion related prompt

❌ Prompt blocked due to unsafe words.


Final Status: Prompt blocked due to unsafe words.
```

Example 2: Girl baking cake in kitchen

```
img, msg = full_pipeline_Testing("Girl baking cake in kitchen ")
print("\nFinal Status:", msg)
```

```
==================================================
🔹 Layer 0: Initial User Input
Prompt: Girl baking cake in kitchen
No fashion related prompt

❌ Prompt blocked due to unsafe words.


Final Status: Prompt blocked due to unsafe words.
```

6.3 Model Performance Observations

The performance of the system was evaluated based on several factors:

- **Image Quality**:
  The Stable Diffusion model with fashion-specific LoRA consistently generated high-quality images with detailed textures, fabric patterns, and appropriate clothing structure. The integration of SwinIR further improved sharpness and resolution.
- **Prompt Understanding**:
  The prompt enhancement layer improved the model's understanding of user inputs by rephrasing and refining vague prompts into more descriptive, fashion-relevant ones. This led to better alignment between user intent and generated output.
- **Safety Filtering**:
  NudeNet effectively detected and filtered out any NSFW or inappropriate content before final output. In test runs, it successfully flagged and removed all unsafe outputs without affecting valid ones.
- **Responsiveness**:
  On Colab/Kaggle (T4 GPU), each image generation cycle took approximately **15–20 seconds**, including enhancement and safety checks — making the system responsive enough for real-time demo or user interaction.
- **Fashion-Only Enforcement**:
  The prompt filtering mechanism accurately identified non-fashion prompts in ~93% of test cases, ensuring domain restriction worked as intended.

6.4 Limitations

Despite the promising results, the system has a few limitations:

- **Basic Prompt Classifier**:
  The current prompt filtering uses keyword-based matching, which may not handle nuanced or ambiguous prompts. More advanced natural language classifiers (e.g., BERT) could improve accuracy.
- **Limited Dataset for Fine-Tuning**:
  While the LoRA weights used are fashion-specific, the generation may still occasionally drift toward generic styles. Fine-tuning with a broader, well-curated fashion dataset could enhance specificity.
- **No Personalization or Virtual Try-On**:
  The current system does not support draping clothes on user-supplied models or real persons. It's limited to generic model generation based on prompts.
- **Dependency on Internet Resources**:
  The system relies on remote APIs and GitHub models (like NudeNet v2), which may face latency or availability issues in offline or restricted environments.
- **Compute Resource Constraints**:
  Running the full pipeline requires a decent GPU (e.g., T4). While Colab/Kaggle provide free access, session timeouts or quota limits may impact usability for extended use.

Chapter 7: Conclusion & Future Work

7.1 Conclusion

This project demonstrates a practical and innovative approach to **fashion-specific text-to-image generation** using a layered architecture combining advanced AI techniques. By integrating prompt enhancement (LLM), fashion-specific image generation (Stable Diffusion with LoRA), and postprocessing layers (SwinIR and NudeNet), the system delivers high-quality and safe fashion visuals aligned with user prompts.

Key contributions include:

- A **domain-restricted generation pipeline** that filters out non-fashion content.
- An **LLM-powered prompt enhancement** module that improves prompt relevance and model interpretability.
- The use of **SwinIR** for super-resolution and **NudeNet** for content safety ensures final outputs are both visually appealing and appropriate.
- A user-friendly **Gradio interface** makes the system accessible to both technical and non-technical users.
- Efficient deployment using free GPUs on **Google Colab/Kaggle**, demonstrating accessibility and reproducibility.

Overall, this project serves as a foundation for building **secure, creative, and controllable AI applications in the fashion domain**, with potential applications in fashion design, e-commerce, and digital marketing.

7.2 Future Enhancements

To further improve and expand this system, several enhancements are proposed:

1. **Advanced Prompt Classification**
   Replace keyword-based filtering with transformer-based models (e.g., BERT or RoBERTa) to better classify fashion-related queries and handle ambiguous prompts.
2. **Fine-Tuning with Curated Fashion Dataset**
   Fine-tune the base Stable Diffusion model with a larger and more diverse fashion dataset to improve accuracy and stylistic variety in generated images.
3. **Virtual Try-On Extension**
   Extend the system to support user-uploaded images for **virtual try-on**, enabling garment draping and personalized fashion previews.
4. **Style Transfer & Editing Tools**
   Integrate fashion-specific style transfer, inpainting, and image editing options for more customization.
5. **Mobile/Web Deployment**
   Optimize and deploy the solution as a web or mobile application with backend support, making it scalable for end-users or businesses.
6. **Real-Time Streaming Output**
   Implement progressive image generation or real-time streaming to provide a more interactive experience for users.
7. **Offline Compatibility**
   Package the system with minimal dependencies and offer an offline mode with quantized models for local execution without internet.

Appendices

This section provides supplementary materials relevant to the project, including code references, example prompts, outputs, and instructions for setting up the development environment.

A. Source Code Snippets

The following are key code snippets from different components of the project:

1. Prompt Filtering Function

```python
def is_prompt_safe(prompt):

    # Convert to lowercase once
    lowered = prompt.lower()
    # First check for unsafe content
    for word in unsafe_keywords:
        if word in lowered:
            return False

    # Then check for fashion relevance
    for word in fashion_keywords:
        if word in lowered:
            return True

    # If no fashion keywords found
    print("No fashion related prompt \n")
    return False
```

2.Prompt Enhancement Using LLM

```python
def enhance_prompt(prompt):
    if not is_prompt_safe(prompt):
        return None, "❌ Unsafe prompt blocked"
    instruction = f"""
        Instruction: Refine the following fashion-related prompt to make it more descriptive and visually clear for a fashio
        1. Ensure the prompt is safe and free from explicit or inappropriate content.
        2. Keep the tone natural and concise (1–2 lines).
        3. Preserve the core idea of the original prompt; avoid altering its main theme or intent.
        Original Prompt: "{prompt}"
        """
    inputs = tokenizer(instruction, return_tensors="pt").to(model.device)
    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_new_tokens=60,do_sample=True,temperature=0.4,top_p=0.9,pad_token_id=tokenizer.eos_token_id
        )
    enhanced = tokenizer.decode(outputs[0], skip_special_tokens=True)
    match = re.search(r'Prompt:\s*"(.*?)"\s*$', enhanced.strip(), re.DOTALL)
    final_prompt = match.group(1) if match else enhanced.split("Prompt:")[-1].strip()
    return final_prompt, "✅ Enhanced Prompt Ready"
```

3. Stable Diffusion Fashion Image Generation

```
# Cell 3: Load Stable Diffusion pipeline
model_id = "stabilityai/stable-diffusion-2-1"

pipe = StableDiffusionPipeline.from_pretrained(
    model_id,
    torch_dtype=torch.float16,
    revision="fp16",
).to("cuda")

# Optional: reduce memory usage
pipe.enable_attention_slicing()
```

B. Sample Prompts and Outputs

Here are a few examples of input prompts, their enhanced versions, and generated outputs:

User Input Prompt	Enhanced Prompt
"red dress with floral design for girls"	"Vibrant red, knee-length dress with whimsical floral patterns and spaghetti straps"
"modern bridal lehenga"	"Elegant bridal lehenga in ivory and gold, with intricate embroidery and traditional jewelry accents"
"man riding a horse"	❌ Rejected – not fashion-related

Screenshots of selected generated images are attached at the end of the document.

C. Dependency Installation and Setup Guide

To run the project locally or in Colab/Kaggle environment, the following setup steps must be followed:

1. Environment Setup

- Python version: >=3.10
- Use virtual environment (optional but recommended)

2. Install Dependencies

pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118

pip install diffusers transformers accelerate

pip install gradio nudenet swinir

3. Load Models

- Stable Diffusion (custom LoRA for fashion)
- SwinIR for image enhancement
- NudeNet for safety filtering
- LLM (e.g., Zephyr 7b or Mistral) for prompt enhancement

4. GPU Setup

- Use **Kaggle Notebooks** with **T4 GPU**
- Ensure session does not time out for long generation loops

References

1. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). *High-Resolution Image Synthesis with Latent Diffusion Models*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10684–10695. https://arxiv.org/abs/2112.10752
2. Hugging Face. (2022). *Transformers Library*. https://github.com/huggingface/transformers
3. Hugging Face. (2022). *Diffusers Library*. https://github.com/huggingface/diffusers
4. Wang, L., Zhang, Y., Liang, D., & Van Gool, L. (2021). *SwinIR: Image Restoration Using Swin Transformer*. https://arxiv.org/abs/2108.10257
5. NudeNet. (2021). *Nude Detection using Deep Learning*. https://github.com/notAI-tech/NudeNet
6. Gradio Team. (2022). *Gradio: Python Library for Creating ML Web Apps*. https://gradio.app/
7. Kaggle. (2022). *Kaggle Notebooks for Deep Learning*. https://www.kaggle.com/code
8. Google Colab. (2022). *Colaboratory: Free Jupyter Notebooks Environment*. https://colab.research.google.com/
9. Zephyr Model (Free, Open-Source Instruction-tuned LLM). Hugging Face - Zephyr: https://huggingface.co/HuggingFaceH4/zephyr-7b-beta
10. GitHub. (n.d.). *ONNX Runtime for NudeNet*. https://github.com/microsoft/onnxruntime