UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II"

DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

CORSO DI LAUREA TRIENNALE

IN

INFORMATICA

Titolo

Sviluppo di una pipeline generativa per la creazione di immagini per il settore della moda

Candidato

Gianluigi Mercorio

N86003834

Relatore

Prof. Porfirio Tramontana

Anno Accademico 2023/2024

Indice

Introduzione

Negli ultimi anni, l'intelligenza artificiale ha registrato progressi significativi rendendo le macchine capaci di simulare processi cognitivi umani, riconoscere schemi complessi e prendere decisioni autonome analizzando grandi quantità di dati.

Le sue applicazioni spaziano in vari settori e sono diverse le modalità con cui si interfaccia alle persone: fornendo assistenza virtuale ad un utente, assistenza nelle diagnosi mediche o assistenza alla guida di veicoli tale da renderli autonomi, dimostrando un potenziale applicativo vasto e rivoluzionario.

L'utilizzo dell'intelligenza artificiale non solo sta migliorando l'efficienza e la precisione in molte attività umane, ma sta anche aprendo nuove frontiere nella creatività e nell'innovazione, trasformando non solo il modo in cui viviamo e lavoriamo ma anche il modo in cui ci poniamo domande e ragioniamo.

L'intelligenza artificiale è una materia che si articola in diverse tecnologie e approcci, ciascuno con specifiche capacità e applicazioni.

Tra questi, quello che sta avendo maggior successo è l'IA generativa, un ramo avanzato che si differenzia dagli altri in quanto focalizzato sulla creazione di contenuti nuovi e originali, piuttosto che sull'analisi e la classificazione dei dati.

Tra i settori in cui l'IA generativa trova applicazione, non poteva mancare quello della moda, storicamente all'avanguardia nell' adottare nuove tecnologie per innovare il design e la produzione.

La presente tesi indaga l'applicazione di modelli di IA generativa nel contesto della moda, sviluppando una *pipeline* che utilizza tali modelli per generare immagini innovative e stilisticamente coerenti al *set* di dati originali su cui vengono addestrati.

La pipeline sviluppata combina modelli addestrati come *Stable Diffusion* e i *LoRA* per creare un sistema robusto e versatile per la generazione di immagini, che si offre come supporto al lavoro dei designer.

Questa tesi non solo dimostra le avanzate capacità tecniche dell'IA nella generazione di immagini, ma esplora anche le implicazioni pratiche e creative di queste tecnologie nel settore della moda.

La *pipeline* generativa sviluppata dimostra come l'integrazione dell'IA generativa stia aprendo nuove frontiere nella creatività digitale, offrendo strumenti potenti per i professionisti del settore e contribuendo all'evoluzione continua del design di moda e dell'attività di marketing che ne deriva.

1. Background

1.1. Intelligenza Artificiale

L'intelligenza artificiale (IA) è una disciplina informatica che sviluppa sistemi capaci di eseguire compiti complessi come il riconoscimento visivo, l'elaborazione del linguaggio naturale, la decisione e la risoluzione di problemi. Utilizzando algoritmi avanzati e grandi quantità di dati, l'IA può apprendere, adattarsi e migliorare nel tempo. Le sue applicazioni spaziano dalla sanità alla finanza, dall'intrattenimento ai trasporti, rivoluzionando vari settori con innovazioni che migliorano l'efficienza e creano nuove opportunità.

« L'intelligenza artificiale (IA) è l'abilità di una macchina di mostrare capacità umane quali il ragionamento, l'apprendimento, la pianificazione e la creatività.

L'intelligenza artificiale permette ai sistemi di capire il proprio ambiente, mettersi in relazione con quello che percepisce e risolvere problemi, e agire verso un obiettivo specifico. Il computer riceve i dati (già preparati o raccolti tramite sensori, come una videocamera), li processa e risponde.

I sistemi di IA sono capaci di adattare il proprio comportamento analizzando gli effetti delle azioni precedenti e lavorando in autonomia. » (Parlamento Europeo, 2023)

1.2. Machine Learning

Il *machine learning* (ML) è un sottocampo dell'intelligenza artificiale che permette ai sistemi di apprendere dai dati forniti in *input* sviluppando le proprie regole per prendere decisioni o fare previsioni.

« Per *machine learning* si intende la scienza in grado di sviluppare algoritmi e modelli statistici utilizzati dai sistemi informatici per lo svolgimento di compiti senza istruzioni esplicite e basandosi, invece, su modelli e inferenza. I sistemi informatici utilizzano gli algoritmi di *machine learning* per elaborare grandi quantità di dati storici e identificare modelli di dati. Ciò permette loro di predire i

risultati in maniera più precisa partendo da un dato *set* di dati iniziali. » (Amazon Web Services, 2024)

1.3. Deep Learning

Il *deep learning*, o apprendimento neurale profondo, è un sottocampo avanzato dell'intelligenza artificiale ed una specializzazione del *machine learning* che utilizza reti neurali artificiali per analizzare e apprendere dai dati. Queste reti, ispirate al cervello umano, sono costituite da molti strati di nodi interconnessi che elaborano informazioni in modo complesso e astratto simulando le modalità di acquisizione delle conoscenze tipiche degli esseri umani.

« Può essere utile immaginare il *deep learning* come un diagramma di flusso con un livello iniziale in cui vengono inseriti dati e uno finale che fornisce risultati. Tra questi due troviamo "livelli nascosti" che elaborano informazioni a diversi livelli, modificando e adeguando il loro comportamento man mano che ricevono nuovi dati. I modelli di *deep learning* possono avere centinaia di livelli nascosti, ognuno dei quali svolge il proprio ruolo nell'analizzare relazioni e schemi all'interno del *set* di dati.

A partire dal livello di inserimento iniziale, composto da più nodi, i dati vengono introdotti nel modello e categorizzati di conseguenza prima di passare al livello successivo. Il percorso seguito dai dati attraverso ogni livello dipende da calcoli stabiliti per ciascun nodo. Infine, i dati passano da un livello all'altro acquisendo osservazioni lungo il percorso, che porteranno in definitiva al risultato, o all'analisi finale, dei dati. » (Red Hat, 2024)

Il *deep learning* è un concetto chiave quando si parla di utilizzare i computer per comprendere il linguaggio umano, la cosiddetta elaborazione del linguaggio naturale (*NLP*) e trova applicazione in vari settori: dalla visione artificiale, con riconoscimento di immagini e video, fino al riconoscimento vocale e alla previsione dei mercati finanziari.

1.4. IA Generativa

L'IA generativa è un campo dell'intelligenza artificiale che si focalizza sulla creazione di nuovi contenuti sfruttando modelli di *deep learning* addestrati su grandi *set* di dati.

« L'intelligenza artificiale generativa (IA generativa) è un tipo di intelligenza artificiale in grado di creare nuovi contenuti e idee, tra cui conversazioni, storie, immagini, video e musica. Le tecnologie di intelligenza artificiale tentano di imitare l'intelligenza umana in attività informatiche non tradizionali come il riconoscimento delle immagini, l'elaborazione del linguaggio naturale (*NLP*) e la traduzione. » (Amazon Web Services, 2024)

L'IA generativa permette di creare contenuti nuovi sottoforma di testi, immagini, video, tracce audio e codice di programmazione.

I principali scenari di utilizzo sono i *chatbot*, ovvero sistemi che interagiscono con gli utenti attraverso il linguaggio naturale per esaminare e una richiesta e fornirgli una risposta, le *pipeline* generative per creazione e l'*editing* di immagini e video, i sistemi di assistenza alla stesura di codice software in maniera automatizzata oppure i sistemi di assistenza alla ricerca scientifica.

L'IA generativa viene utilizzata quindi in contesti professionali per svolgere in maniera efficiente attività altrimenti dispendiose in termini di tempo dimostrandosi uno strumento utile ed efficace per i professionisti nello svolgimento delle loro mansioni.

1.5. Modelli di base

Per modelli di base si i intendono i modelli di *deep learning* in grado di svolgere una varietà di compiti senza essere specificamente progettati per ciascuno di essi.

Questi modelli utilizzano architetture avanzate e sono addestrati su enormi volumi di dati generici e diversificati in modo da acquisire conoscenze generali che possono essere applicate a molteplici contesti.

Una volta completato l'addestramento, è possibile perfezionare il modello di base per scenari di utilizzo specifici. Come denota il nome, si tratta di modelli che possono fare da base per moltissime applicazioni.

Tra le tipologie di modelli di base troviamo:

- *Large Language Models (LLMs)*, pre-addestrati su enormi quantità di testo per comprendere e generare linguaggio naturale;
- Modelli di Visione Artificiale, progettati per compiti di riconoscimento e classificazione delle immagini;
- Modelli di Serie Temporali, utilizzati per analizzare dati sequenziali e fare previsioni basate su serie temporali;
- Modelli di Diffusione, particolarmente efficaci nella generazione di immagini dettagliate e di alta qualità, partendo da descrizioni testuali o altri *input*.

Considerando il costo e gli sforzi necessari per l'addestramento dei modelli di base da zero, è pratica comune acquisire modelli già addestrati e occuparsi poi della loro personalizzazione. Le tecniche per la personalizzazione dei modelli di base sono diverse e una delle più diffuse è la tecnica di *fine tuning.*

1.6. Fine tuning

Il *fine tuning* è il processo di perfezionamento dei modelli di base per la creazione di un nuovo modello più adatto a specifici compiti o domini. Tale approccio offre maggiore efficienza per determinati scenari di utilizzo rispetto all'uso di un modello generico.

In genere il *fine tuning* richiede un numero di dati e di ore di lavoro nettamente inferiore rispetto all'addestramento iniziale del modello di base. Se per l'addestramento iniziale ci possono volere settimane o persino mesi, il processo di *fine tuning* si può concludere anche in poche ore.

Chi sceglie di utilizzare un modello generico deve inserire specifici esempi e istruzioni articolate per ottenere lo stesso risultato che un modello raffinato con *fine tuning* riesce a produrre con richieste più semplici.

Con l'attività di *fine tuning*, quindi, oltre ad aumentare la precisione e la specificità del modello, si risparmiano tempo e risorse.

1.7. Modelli di Diffusione

I Modelli di Diffusione sono una tipologia di modelli di base che apprendono il processo di conversione di un'immagine in rumore visivo ed invertono poi il processo per generare un'immagine, partendo dal solo rumore e raffinandolo fino a ottenere un'immagine realistica.

« I modelli di diffusione creano nuovi dati apportando iterativamente modifiche casuali controllate a un campione di dati iniziale. Iniziano con i dati originali e aggiungono leggere modifiche (rumore), rendendoli progressivamente meno simili all'originale. Questo rumore viene controllato attentamente per garantire che i dati generati rimangano coerenti e realistici.

Dopo aver aggiunto rumore in diverse iterazioni, il modello di diffusione inverte il processo. La riduzione del rumore inversa rimuove gradualmente il rumore per produrre un nuovo campione di dati simile all'originale. » (Amazon Web Services, 2024)

1.8. Pipeline generative

Le *pipeline* generative sono processi strutturati che utilizzano modelli di intelligenza artificiale per creare nuovi contenuti a partire da *input* specifici.

Per la creazione di una *pipeline* vengono coinvolti vari passaggi, come la preparazione dei dati, l'addestramento del modello generativo che si intende utilizzare, la generazione del contenuto richiesto e l'ottimizzazione finale dei contenuti prodotti.

L'obiettivo principale di una *pipeline* generativa è produrre *output* di alta qualità che siano coerenti con le specifiche richieste dell'utente, automatizzando e ottimizzando il processo creativo.

Per lo sviluppo di una *pipeline* generativa per la creazione di immagini legate al mondo della moda si inizia con la raccolta di dati visivi dei capi, seguita dall'addestramento di un modello generativo e termina con la produzione di nuove immagini di design.

Affinché il processo generativo possa avere luogo, deve essere definito un flusso di elaborazione che costituisce la struttura principale della *pipeline*; devono essere individuati dei modelli generativi da inserire in tale struttura e deve essere indicata la richiesta testuale da fornire come *input*, chiamata *prompt*.

Attraverso l'utilizzo di specifici modelli di IA, il risultato ottenuto dalla generazione può essere elaborato ulteriormente per migliorarne alcuni aspetti, come dettagli visivi o la risoluzione.

1.9. Modelli LoRA

I modelli *LoRA* (*Low-Rank Adaptation*) sono reti neurali generative ottenute come risultato di attività di *fine tuning*, addestrando il modello di base su un particolare tipo di immagini.

Tali modelli risultano estremamente efficienti nei casi in cui si voglia generare immagini che includono stili artistici particolari e contenuti realmente esistenti come i volti di persone o particolari capi d'abbigliamento.

Nelle *pipeline* generative si possono combinare modelli di diffusione e i modelli *LoRA* da essi ottenuti per creare nuovi contenuti visivi. Tale approccio prevede che il modello di base guidi il processo di generazione, incorporando le conoscenze specifiche apprese dai *LoRA*. In questo modo, è possibile generare immagini che contengono elementi e stili specifici, altresì difficili o talvolta impossibili da ottenere, aumentando complessivamente la creatività e le capacità generative della *pipeline*.

2. Pipeline generativa

2.1. Introduzione

Si vuole sviluppare una *pipeline* generativa che, combinando modelli di IA all'interno di un flusso di elaborazione, permetta la generazione di immagini secondo le richieste di un utente, identificate da *input* testuali detti *prompt*.

Per la realizzazione di una *pipeline* di questo tipo si è scelto di utilizzare il software *ComfyUI*, che offre un ambiente grafico per la loro creazione ed esecuzione, il modello generativo di base *StableDiffusion* ed ulteriori modelli *LoRA* ottenuti come risultato di attività di *fine tuning* a partire da quest'ultimo.

2.2. Tecnologie utilizzate

2.2.1. Stable Diffusion

Stable Diffusion è un modello di Diffusione che produce immagini, video e animazioni a partire da un *prompt* testuale.

Si differenzia dagli altri modelli per l'utilizzo dello spazio latente per codificare un'immagine.

Lo spazio latente rappresenta una sorta di mappa in cui i dati complessi, come immagini o testi, vengono compressi in rappresentazioni più semplici e dense durante il processo di codifica. Ogni punto nello spazio latente corrisponde a una versione compressa dei dati originali e può essere decodificato per generare nuovi dati simili agli originali.

La sua implicazione riduce notevolmente i requisiti di elaborazione e rende *Stable Diffusion* meno oneroso rispetto agli altri modelli generativi, mantenendo un'alta qualità generativa.

Per ricreare un'immagine dallo spazio latente, *Stable Diffusion* utilizza:

- Un *autoencoder* variazionale per manipolare le immagini codificate in uno spazio latente e da ricodificare in pixel.
- La diffusione inversa, un processo che iterativamente rimuove il rumore da un'immagine fino ad ottenerne una chiara e dettagliata.
- Un predittore di rumore, fondamentale per consentire il processo di diffusione inversa. Esso stima la quantità di rumore nello spazio latente e la sottrae dall'immagine.

In sintesi, *Stable Diffusion* genera immagini dallo spazio latente attraverso un processo iterativo di *denoising* (riduzione del ruomore), dove l'immagine viene raffinata passo dopo passo, partendo dal rumore completo fino a raggiungere una forma chiara e dettagliata.

2.2.2. ComfyUI

ComfyUI è un software *open-source* sviluppato da *comfyanonimous* per facilitare lo sviluppo e l'esecuzione di *pipeline* complesse che generano contenuti visivi a partire da modelli di IA generativa.

L'interfaccia grafica consente agli utenti di progettare e personalizzare le proprie *pipeline* utilizzando una varietà di modelli di IA e tecniche avanzate, consentendo agli utenti di aggiungere componenti, detti nodi, e connetterli tra loro per definire il flusso di elaborazione dei dati.

All'interno della *pipeline* è possibile aggiungere:

- Modelli di Diffusione e modelli *LoRA*, per la generazione dei contenuti visivi a partire da input testuali ed immagini latenti.
- Modelli *Variational Autoencoder* (*VAE*), per la codifica di dati nello spazio latente e la loro decodifica.
- Nodi *Sampler*, per ridurre il rumore dalle immagini latenti;
- Modelli di *Upscaling*, per migliorare la risoluzione e la qualità dei contenuti generati;

- Nodi *CLIP*, per la scrittura di *prompt* testuali che vengono codificati e utilizzati come *input* per condizionare il processo generativo. Rappresentano la richiesta che il modello deve soddisfare.

ComfyUI offre, inoltre, la possibilità di salvare, caricare e modificare le *pipeline* generative create dagli utenti per favorire la riutilizzabilità e la personalizzazione. Tale *feature* è realizzata codificando ciascuna *pipeline* in un *workflow*, ovvero un *file JSON* che contiene tutte le informazioni strutturali.

Fornisce un *API* che consente l'interazione con la *pipeline* attraverso l'indicazione del *server address*, cioè la combinazione di *URL* e numero di porta su cui il software è in ascolto, e l'indicazione del *workflow*, quindi l'effettiva struttura della *pipeline* da eseguire.

Rappresenta, quindi, un potente strumento per esplorare e sfruttare il potenziale dei modelli generativi.

Con la sua interfaccia grafica intuitiva, le numerose funzionalità e la varietà di nodi a disposizione, risulta un'opzione ideale per sviluppatori, designer e ricercatori interessati alla sperimentazione nel campo della generazione automatica di contenuti visivi.

2.3. Modelli utilizzati

Per lo sviluppo della *pipeline* sono stati utilizzati i seguenti modelli di IA:

- modello di Diffusione ***StableDiffusion_XL_1.0_base***[1], come modello di generazione di base;
- modello di Diffusione ***StableDiffusion_XL_1.0_refiner***[2], in combinazione con la versione base per fornire maggiori livelli di dettaglio e coerenza generativa;

Di seguito viene riportato il grafico delle *performance*[3] che fornisce un quadro completo sulle prestazioni dei vari modelli di *StableDiffusion* e da cui si evince che l'utilizzo combinato dei modelli *base* e *refiner* della versione *SDXL_1.0* sia la soluzione migliore tra quelle disponibili.

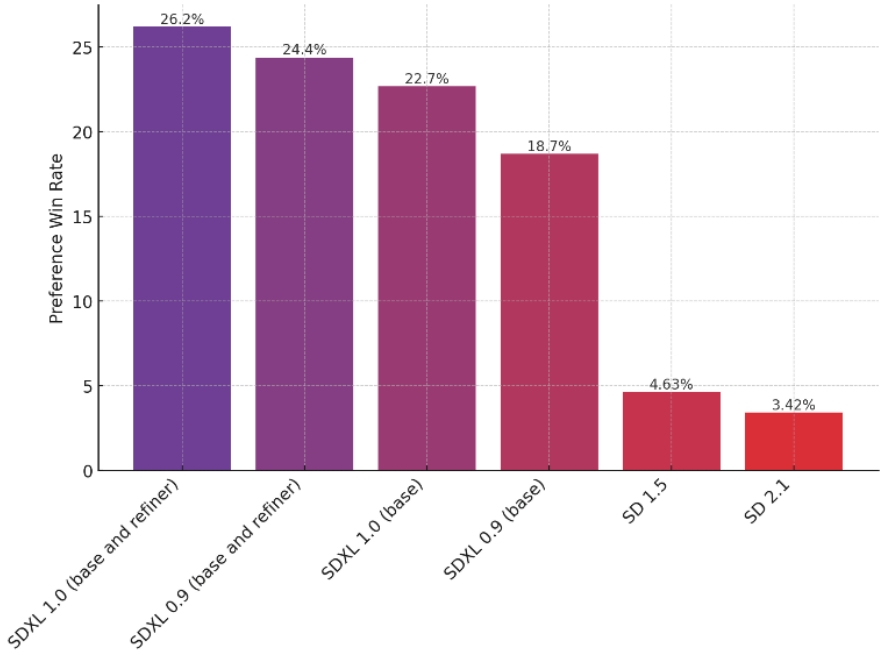

Figura 1: Grafico performance modelli StableDiffusion

[1] sviluppato da *StabilityAI* e disponibile su *huggingface.co* .
[2] sviluppato da *StabilityAI* e disponibile su *huggingface.co* .
[3] pubblicato su *HuggingFace.com* da *StabilityAI.*

- modello *LoRA* ***Adidas_V2_SDXL_02***[4]*;*

Figura 2: Capacità generativa LoRA Adidas_V2_SDXL_02

- modello *LoRA* ***arman1jacket-xl-step00001500***[5]*;*

Figura 3: Capacità generativa LoRA arman1jacket-xl-step00001500

- modello *LoRA* ***Parkasite_XL-000010***[6]*;*

Figura 4: Capacità generativa LoRA Parkasite_XL-000010

[4] sviluppato da *denrakeiw* e disponibile su *civitai.com* .
[5] sviluppato da *Steve Norman* e disponibile su *punya.ai* .
[6] sviluppato da *Clara1* e disponibile su *civitai.com* .

- modello ***StableDiffusion_VAE***[7], per la decodifica delle immagini latenti generate con il modello *StableDiffusion_XL* (valido per entrambe le versioni)

- modello ***Upscaler 4xNMKDSuperscale_4xNMKDSuperscale***[8], per aumentare la risoluzione delle immagini generate.

[7] sviluppato da *StabilityAI* e disponibile su *huggingface.co* .
[8] sviluppato da *Samael1976* e disponibile su *civitai.com* .

2.4. Struttura

La *pipeline* sviluppata si articola secondo tale struttura:

- due nodi ***Load Checkpoint***, per il caricamento dei modelli di Diffusione utilizzati;
- tre nodi ***Load LoRA***, per il caricamento dei modelli *LoRA* utilizzati, da collegare al modello di Diffusione da cui sono stati ottenuti;
- quattro nodi ***CLIP Text Encode***, per la definizione dei *prompt* testuali da fornire in *input* ai *Sampler* per condizionare la generazione.
- un nodo ***Empty Latent Image***, per generare un'immagine vuota, già codificata nello spazio latente*;*
- due nodi ***KSampler***, che orchestrano la generazione dell'immagine secondo specifici parametri*;*
- un nodo ***Load VAE***, per il caricamento del modello *VAE* utilizzato;
- un nodo ***VAE Decode***, per la decodifica dell'immagine generata nello spazio latente utilizzando il modello *VAE* caricato;
- un nodo ***Load Upscale Model***, per il caricamento del modello *Upscaler* utilizzato;
- un nodo ***Upscale Image***, per effettuare l'*upscaling* dell'immagine generata utilizzando il modello *Upscaler* caricato;
- un nodo ***Save Image***, per visualizzare e salvare l'immagine generata.

Vengono di seguito riportate immagini descrittive della struttura presentata:

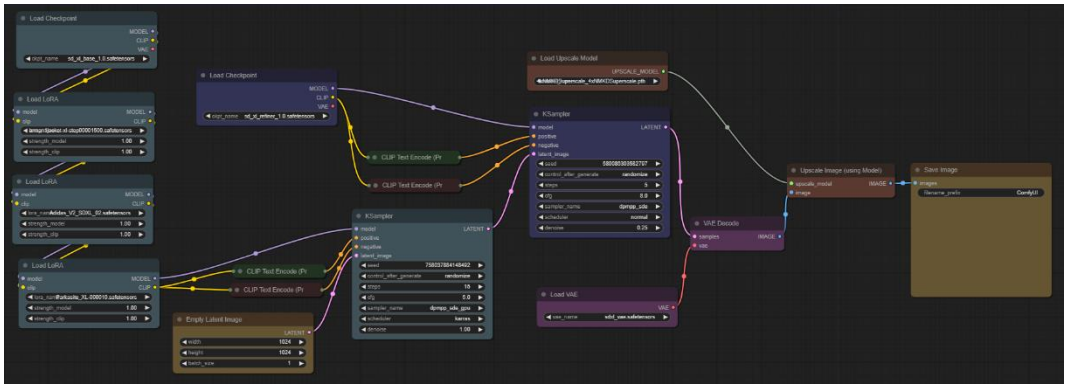

Figura 5: Pipeline completa

Figura 6: Nodo Load Checkpoint - SDXL_Base

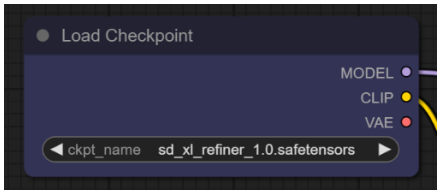

Figura 7: Nodo Load Checkpoint - SDXL_Refiner

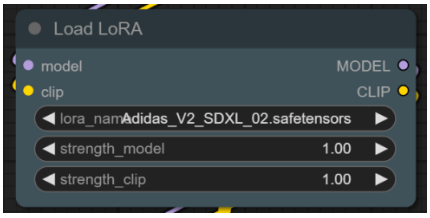

Figura 8: Nodo Load LoRA - Adidas

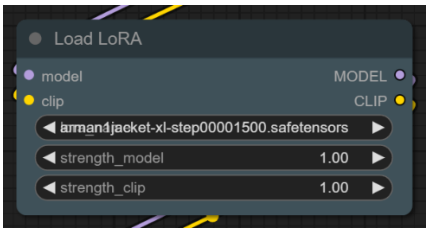

Figura 9: Nodo Load LoRA - Armani

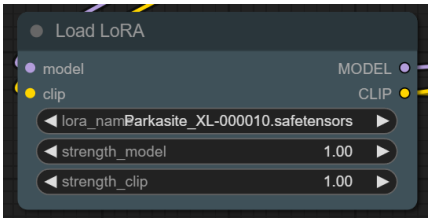

Figura 10: Nodo Load LoRA - Parkasite

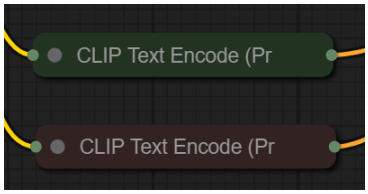

Figura 11: Nodi CLIP Text Encode

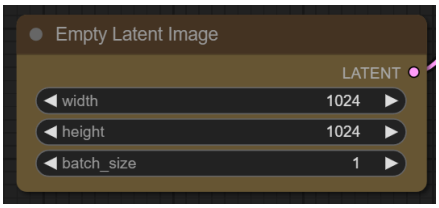

Figura 12: Nodo Empty Latent Image

Figura 13: Nodo KSampler - SDXL_Base

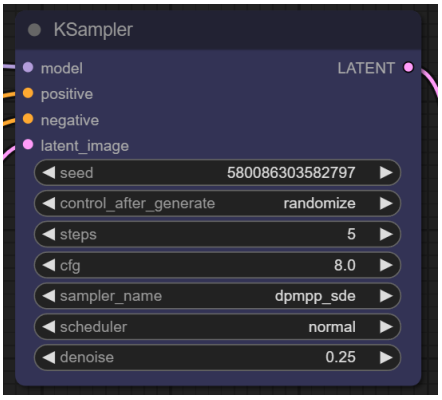

Figura 14: Nodo KSampler - SDXL_Refiner

Figura 15: Nodo Load VAE

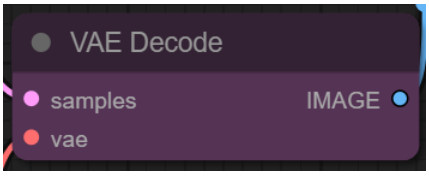

Figura 16: Nodo VAE Decode

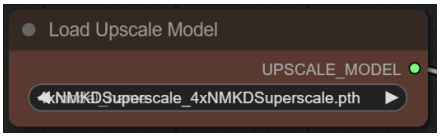

Figura 17: Nodo Load Upscale Model

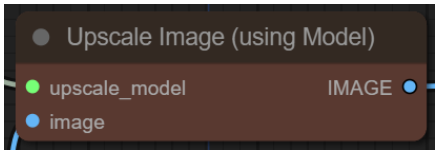

Figura 18: Nodo Upscale Image

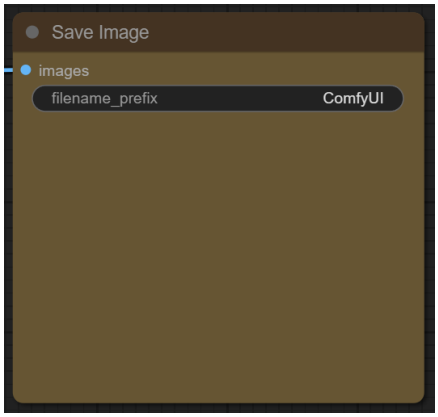

Figura 19: Nodo Save Image

3. Genai software

3.1. Introduzione

Si vuole realizzare un applicativo software il cui scopo è quello di offrire uno strumento che consenta di generare immagini in modo semplice e intuitivo.

Il software si presenta come singola pagina web che permette agli utenti di interagire tramite un'interfaccia grafica (*GUI*) per inserire *prompt* testuali, visualizzare e selezionare capi d'abbigliamento generabili, scaricare i risultati e visualizzare esempi di generazione.

Inoltre, permette di visualizzare e aggiornare le impostazioni del software, come l'indirizzo del server della *pipeline* e il *workflow* corrente.

Il software si avvale dell'*API* fornita da *ComfyUI* per inviare le richieste alla *pipeline* generativa, sulla base dei dati forniti dall'utente, e raccogliere i risultati del flusso di elaborazione mostrandoli all'utente come *output*.

3.2. Requisiti

3.2.1. Requisiti Funzionali

Vengono di seguito elencati i requisiti funzionali, ovvero tutti i servizi (o funzioni) che il sistema deve offrire.

1. **Inserimento dei prompt**: L'utente deve poter inserire un *prompt* testuale positivo e un *prompt* testuale negativo per specificare rispettivamente cosa vuole e cosa non vuole generare.
2. **Visualizzazione dei capi d'abbigliamento**: L'utente deve poter visualizzare e selezionare i capi d'abbigliamento che la *pipeline* è in grado di generare di *default*, mostrando l'identificativo associato.
3. **Interazione con la *pipeline***: L'utente deve poter inoltrare alla *pipeline* la propria richiesta, definita con l'inserimento dei *prompt* e l'eventuale selezione del capo d'abbigliamento.

4. **Visualizzazione e download del risultato**: L'utente deve poter visualizzare e scaricare il risultato ottenuto dalla *pipeline* generativa.
5. **Visualizzazione degli esempi di generazione**: L'utente deve poter visualizzare esempi di immagini generate e i *prompt* corrispondenti.
6. **Visualizzazione delle impostazioni correnti**: L'utente deve poter visualizzare le impostazioni correnti del software, come l'indirizzo del server della *pipeline*, il *workflow* corrente e le posizioni dei *prompt* all'interno di quest'ultimo.
7. **Aggiornamento delle impostazioni**: L'utente deve poter modificare le impostazioni, inserendo un indirizzo del server personalizzato, caricando un *workflow* personale oppure specificando le posizioni dei *prompt* al suo interno.
8. **Visualizzazione delle impostazioni di default**: L'utente deve poter visualizzare l'indirizzo del server standard – per un utilizzo in locale – e il *workflow* di *default*, ovvero quello corrispondente alla *pipeline* fornita.

3.2.2. Requisiti Non Funzionali

Vengono ora elencati i requisiti non funzionali ovvero tutti i vincoli sui servizi offerti dal sistema.

1. **Usabilità**: L'interfaccia utente deve essere intuitiva e facile da usare, consentendo agli utenti di completare le operazioni con pochi clic.
2. **Affidabilità**: Il sistema deve essere stabile e gestire correttamente errori e *input* non validi, assicurando che i dati elaborati non vengano persi o corrotti.
3. **Flessibilità**: Il sistema deve poter essere usato con configurazioni personalizzate.
4. **Portabilità**: Il software deve essere eseguibile su diverse piattaforme, inclusi *Windows*, *macOS* e *Linux*, sfruttando la portabilità del linguaggio *Python*.

3.3. Analisi dell'Architettura

Il software possiede un'architettura monolitica articolata in sei moduli principali.

1. **Interfaccia Utente (*GUI*)**: modulo che permette l'interazione dell'utente con il sistema, realizzato utilizzando la libreria *Gradio*.
2. **Gestore degli esempi**: modulo che si occupa di fornire al modulo della *GUI* gli esempi generativi da mostrare all'utente.
3. **Gestore della galleria**: modulo che si occupa di fornire al modulo della *GUI* i capi d'abbigliamento che la *pipeline* fornita è capace di generare.
4. **Gestore del *workflow***: modulo che gestisce l'aggiornamento del *workflow* con i dati inseriti dall'utente. Il *workflow* aggiornato costituisce il corpo della richiesta di elaborazione.
5. **Comunicatore con *ComfyUI***: modulo che utilizza l'*API* di *ComfyUI* per inoltrare alla *pipeline* le richieste basate sul *workflow* aggiornato con i dati inseriti dall'utente. Il modulo si occupa, inoltre, di ricevere il risultato del flusso di elaborazione.
6. **Gestore delle impostazioni**: modulo che consente di visualizzare e aggiornare le impostazioni correnti del software, inclusi *server address*, *workflow* e le posizioni dei *prompt* associati.

3.4. Esempi d’uso

3.4.1. Schermata principale

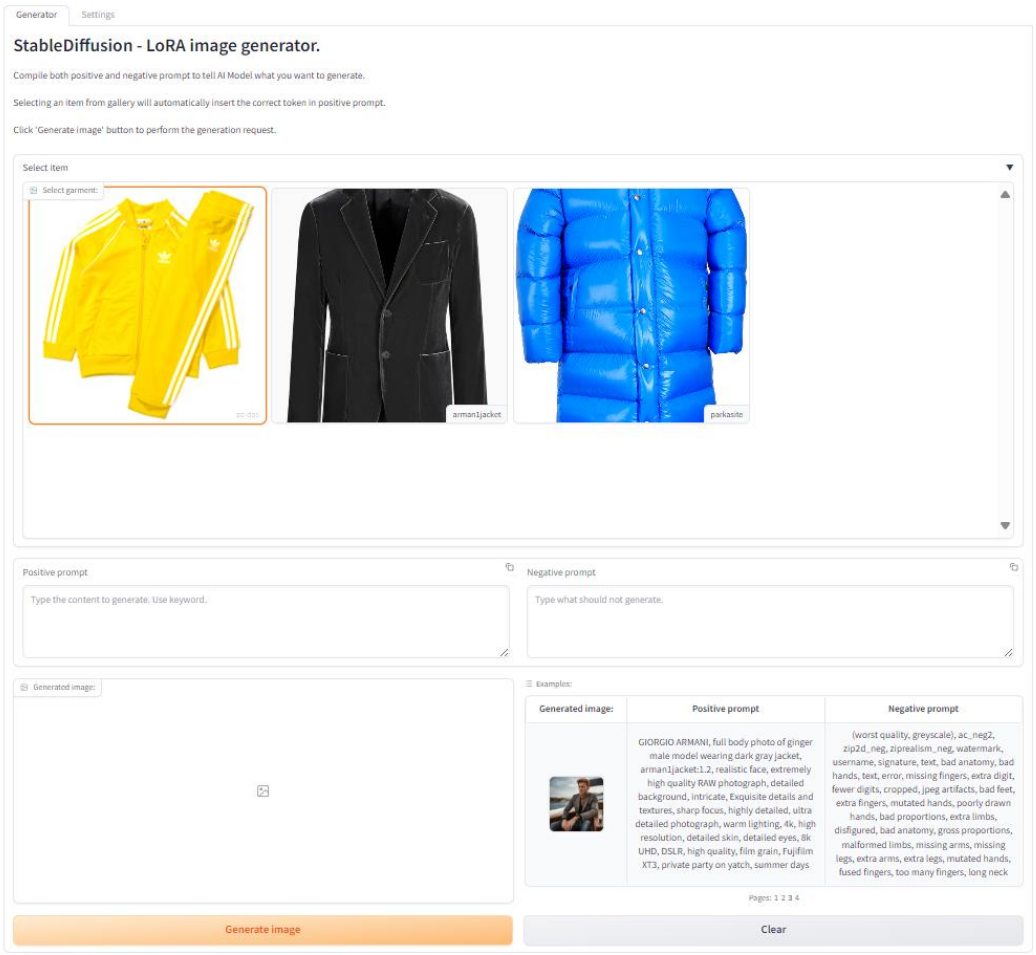

Figura 20: Genai software - schermata principale

3.4.2. Schermata visualizzazione esempi

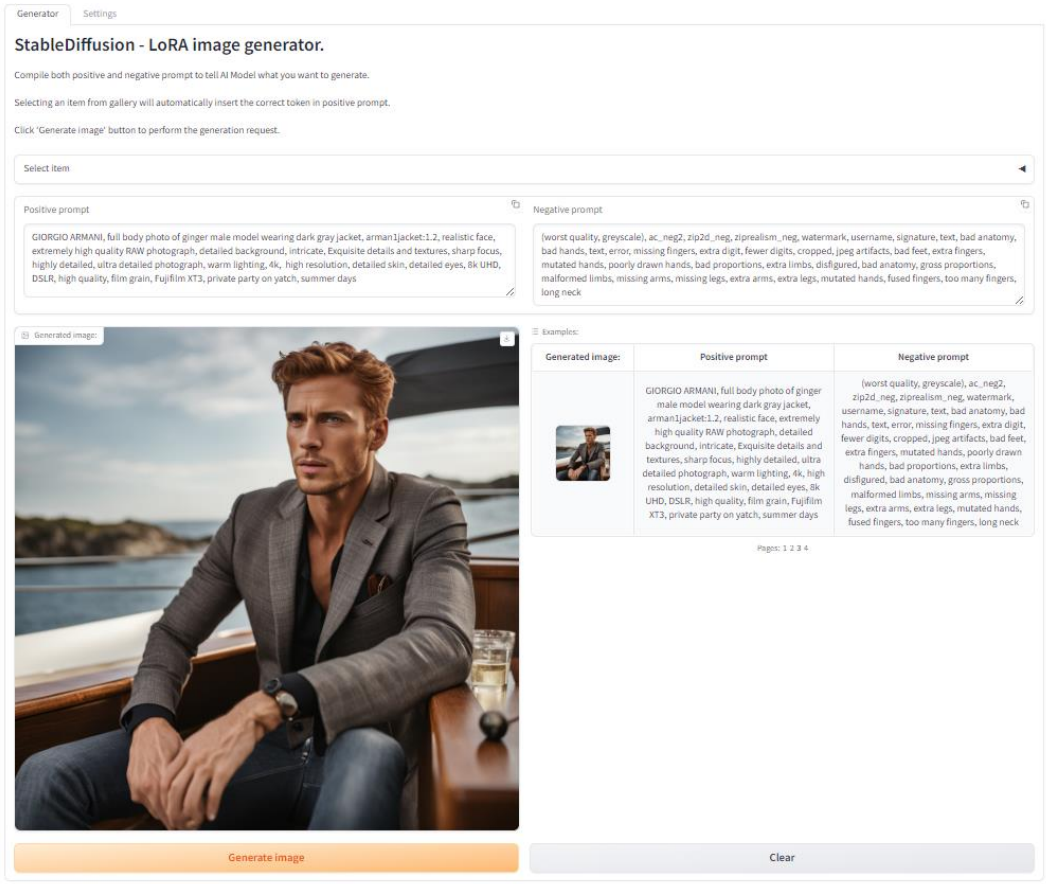

Figura 21: Genai software - schermata visualizzazione esempi

3.4.2. Schermata gestione impostazioni

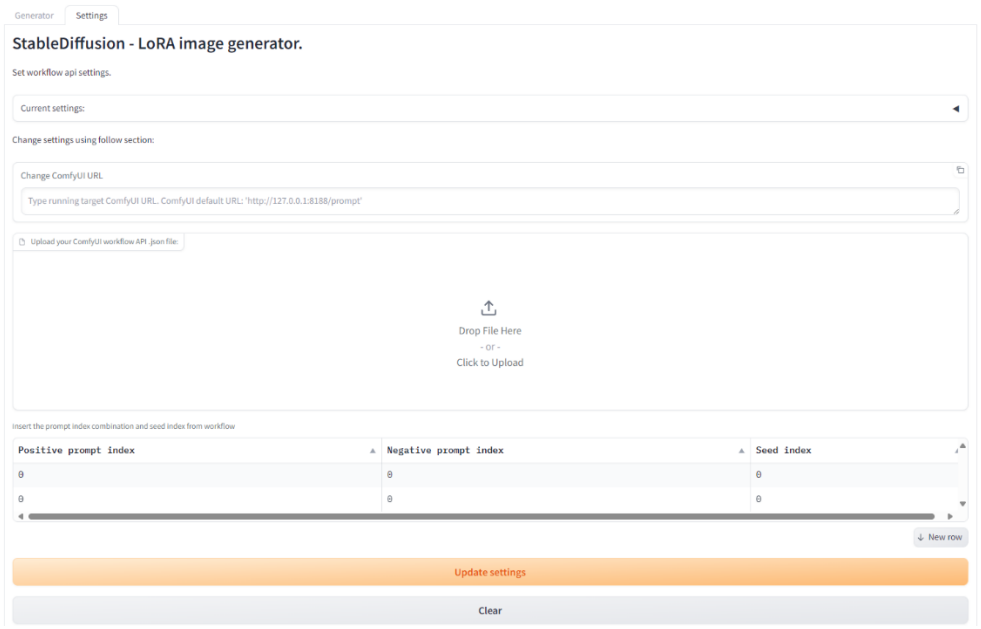

Figura 22: Genai software - schermata gestione impostazioni

3.4.3. Schermata impostazioni correnti

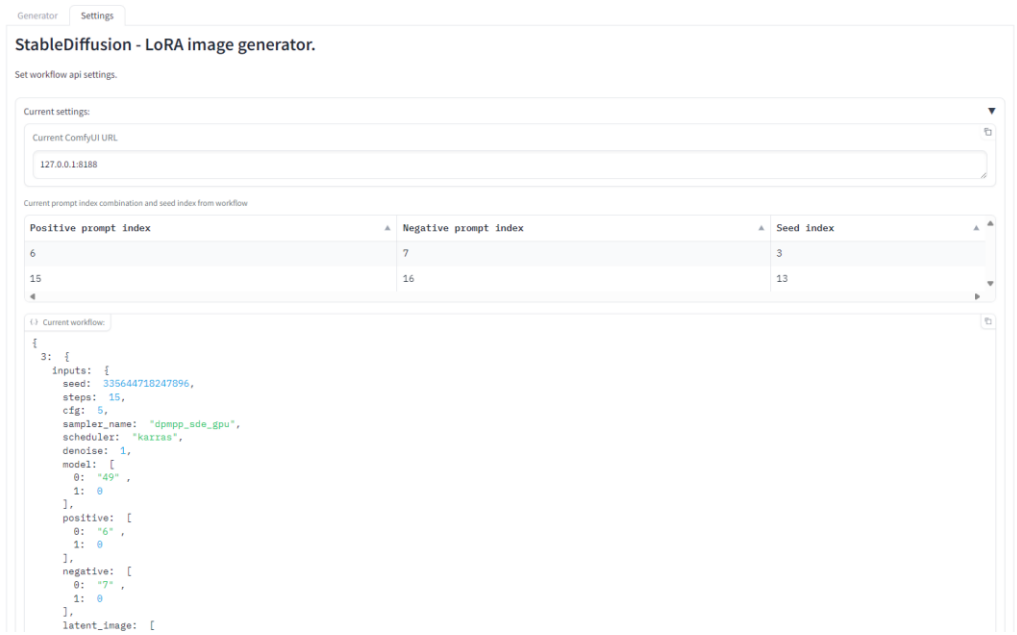

Figura 23: Genai software - schermata impostazioni correnti

4. Genai dataset generator software

4.1. Introduzione

Si vuole realizzare un applicativo software il cui scopo è quello di offrire uno strumento che consenta di generare in modo rapido ed automatizzato un *dataset* di immagini finalizzato all'addestramento di modelli di intelligenza artificiale.

Il software si presenta come singola pagina *web* che permette agli utenti di interagire tramite un'interfaccia grafica (*GUI*) per caricare una cartella di immagini da elaborare.

L'applicativo consente di effettuare operazioni sulle immagini come il ridimensionamento, la rinominazione enumerata sulla base di un identificativo personale (*token*), l'etichettatura delle immagini in maniera manuale e automatizzata; tale operazione prevede un'elaborazione da parte di un modello di IA di tipo *text-to-image,* che genera una descrizione testuale del contenuto visivo dell'immagine processata.

I risultati ottenuti vengono tabulati e rilasciati all'utente come archivio compresso.

4.2. Requisiti

4.2.1. Requisiti Funzionali

Vengono di seguito elencati i requisiti funzionali, ovvero tutti i servizi (o funzioni) che il sistema deve offrire.

1. **Caricamento delle immagini**: L'utente deve poter caricare una cartella contenente immagini. Questo passaggio è obbligatorio per l'elaborazione successiva.
2. **Ridimensionamento delle immagini**: L'utente deve poter scegliere di ridimensionare le immagini caricate specificando, in *pixel*, le dimensioni desiderate. Questa operazione è facoltativa.

3. **Rinominazione delle immagini**: È obbligatorio per l'utente rinominare le immagini caricate specificando un *token*, ovvero un nome significativo che identifichi ciascuna immagine. Per ciascuna immagine, al *token* dovrà essere associato un numero seriale per distinguerle.
4. **Etichettatura manuale**: L'utente deve poter associare a ciascun'immagine una descrizione testuale personalizzata.
5. **Etichettatura automatizzata**: L'utente deve poter utilizzare un modello IA per estrapolare una descrizione testuale del contenuto visivo delle immagini in maniera automatizzata.
6. **Tabulazione dei risultati**: Alla fine del processo i dati relativi ai risultati ottenuti devono organizzati in maniera tabulare.
7. **Acquisizione dei risultati**: L'utente deve poter scaricare un archivio compresso in cui sono contenute le risorse di interesse: le immagini processate, i file di testo generati e il file tabulare in cui devono essere organizzati i dati dei risultati ottenuti.

4.2.2. Requisiti Non Funzionali

Vengono ora elencati i requisiti non funzionali ovvero tutti i vincoli sui servizi offerti dal sistema.

1. **Usabilità**: L'interfaccia utente deve essere intuitiva e facile da usare, consentendo agli utenti di completare le operazioni con pochi clic**.**
2. **Prestazioni**: Il sistema deve elaborare un numero considerevole di immagini in tempi ragionevoli, garantendo un'esperienza fluida.
3. **Affidabilità**: Il sistema deve essere stabile e gestire correttamente errori e *input* non validi, assicurando che i dati elaborati non vengano persi o corrotti.
4. **Portabilità**: Il software deve essere eseguibile su diverse piattaforme, inclusi *Windows*, *macOS* e *Linux*, sfruttando la portabilità del linguaggio *Python*.

4.3. Analisi dell'Architettura

Il software possiede un'architettura monolitica articolata in nove moduli principali.

1. **Interfaccia Utente (*GUI*)**: modulo che permette l'interazione dell'utente con il sistema, realizzato utilizzando un la libreria *Gradio*.
2. **Gestore dei file**: modulo che si occupa del salvataggio e dell'organizzazione delle immagini caricate e delle risorse sviluppate.
3. **Ridimensionatore**: modulo che si occupa di ridimensionare le immagini, realizzato utilizzando la libreria *OpenCV*.
4. **Rinominatore**: modulo che gestisce la rinominazione enumerata di immagini e file di testo associati.
5. **Modulo IA**: modulo che si occupa del caricamento e dell'interazione con il modello IA utilizzato per l'etichettatura automatizzata delle immagini. Realizzato utilizzando il modello IA *BlipImageCaptioning*[9] e la libreria *transformers*[10].
6. **Etichettatore**: modulo che si occupa della creazione dei file testuali associati a ciascuna immagine, etichettandole con il *token* fornito, la descrizione specificata dall'utente e, se previsto, la descrizione generata dal modulo IA.
7. **Tabulatore**: modulo che si occupa della tabulazione dei risultati ottenuti generando un file corrispondente.
8. **Archiviatore**: modulo che si occupa di comprimere in una cartella le risorse sviluppate dal flusso di elaborazione.
9. **Preprocessore**: modulo che si occupa di controllare la presenza del modello IA e l'inizializzazione delle risorse necessarie al flusso di elaborazione.

[9] sviluppato da *Salesforce* e fruibile in maniera *open source* su *Huggingface.com* .
[10] sviluppata e fruibile in maniera *open source* su *Huggingface.com* .

4.4. Esempi d’uso

4.4.1. Schermata principale

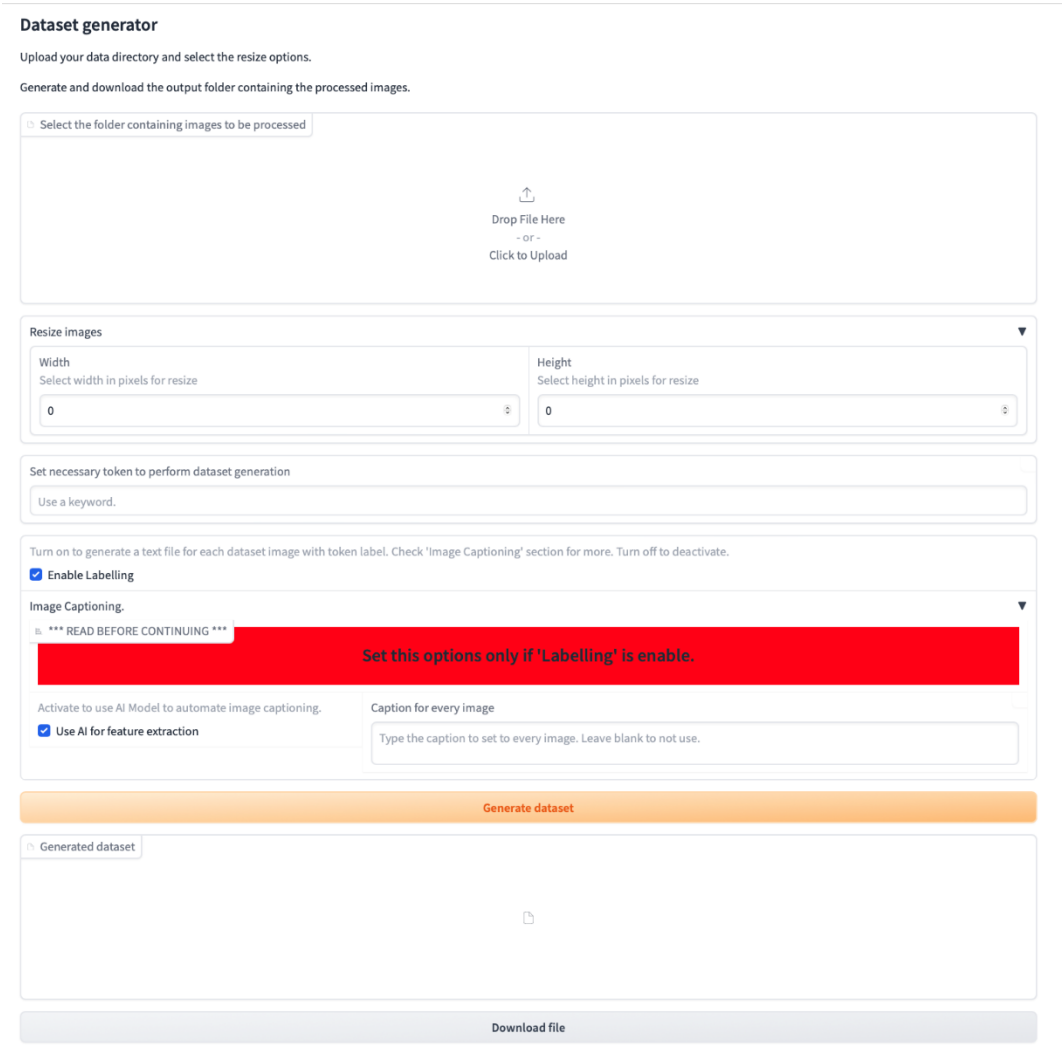

Figura 24: Genai dataset generator software - schermata principale

5. Validazione

5.1. Introduzione

La validazione di una *pipeline* generativa è un processo cruciale per garantire che le immagini prodotte soddisfino i criteri di qualità e coerenza definiti dagli utenti.

In questo capitolo, viene descritto l'approccio qualitativo adottato per la validazione della *pipeline* generativa sviluppata con *ComfyUI*.

L'obiettivo è misurare la qualità e la coerenza tematica delle immagini generate rispetto alle combinazioni *prompt* positivi e negativi forniti come *input* per il processo di generazione.

5.2. Obiettivi della validazione

Di seguito vengono elencati gli obiettivi della validazione.

- **Valutare la coerenza tematica**: Assicurare che le immagini generate siano coerenti con i temi e gli elementi specificati nei *prompt* positivi.
- **Verificare l'aderenza ai prompt negativi**: Garantire che gli elementi e i temi specificati nei *prompt* negativi siano assenti nelle immagini generate.
- **Misurare la qualità visiva**: Valutare la qualità estetica delle immagini prodotte, assicurando che siano visivamente piacevoli e prive di artefatti (anomalie visive, imperfezioni).

5.3. Metodologia di validazione

Di seguito vengono presentati i passaggi necessari alla validazione qualitativa della pipeline generativa.

1. **Selezione del campione di immagini:** Generare un insieme di immagini utilizzando una varietà di *prompt* positivi e negativi, mirando a coprire una gamma ampia di temi e contenuti.

2. **Valutazione da parte degli utenti:** Coinvolgere un gruppo di utenti che esamineranno le immagini generate e assegneranno un punteggio in base a dei criteri stabiliti.
3. **Raccolta dei *feedback*:** Raccogliere i punteggi degli utenti assegnati a ciascuna immagine.
4. **Calcolo dei risultati:** Analizzare i dati e calcolare la media dei punteggi per ogni criterio di valutazione.
5. **Analisi dei risultati:** Confrontare i risultati con gli obiettivi iniziali per determinare il successo della pipeline generativa.

5.4. Criteri di valutazione

Di seguito vengono riportati i criteri di valutazione definiti:

a) **Coerenza tematica**: Ogni immagine viene valutata per quanto riguarda la presenza e l'accuratezza degli elementi tematici indicati nei *prompt* positivi.
b) **Assenza di elementi indesiderati**: Si verifica che le immagini non contengano elementi descritti nei *prompt* negativi.
c) **Qualità visiva**: Gli utenti valutano la nitidezza, l'assenza di artefatti (imperfezioni, anomalie visive) e la qualità estetica generale delle immagini.

5.5. Validazione effettuata

Di seguito viene presentata l'attività di validazione effettuata secondo le metodologie e i criteri di valutazione presentati.

5.5.1. Descrizione dell'attività

L'attività di validazione svolta ha coinvolto un gruppo di 5 utenti con competenze specifiche nel settore della moda, che hanno assegnato alle immagini proposte un punteggio compreso tra 1 e 5 per ciascun criterio di valutazione (coerenza tematica, assenza di elementi indesiderati, qualità visiva).

La *pipeline* sviluppata consente di generare immagini contenenti ben tre capi d'abbigliamento specifici diversi ed offre la possibilità di generare tali capi con colori diversi. Pertanto, affinché l'attività possa essere ritenuta maggiormente affidabile, si è scelto di fornire agli utenti un campione di immagini che contenesse i capi d'abbigliamento disponibili, generati secondo colorazioni diverse.

5.5.2. Campione di immagini

Il campione di immagini fornito agli utenti è stato generato come segue:

Per ciascuno dei 3 capi d'abbigliamento sono stati definite 5 combinazioni di *prompt* diversi.

Ciascuna combinazione è stata arricchita da 2 tipi di colorazioni diverse, ottenendo 10 combinazioni diverse per ciascun capo d'abbigliamento.

Si è tenuto conto della *stocasticità*, ovvero la proprietà dei modelli generativi di fornire risultati diversi processando la stessa combinazione di *prompt*. Pertanto, si è scelto di processare 2 volte ciascuna delle combinazioni di *prompt* individuate, generando un totale di 20 immagini per capo d'abbigliamento.

Il campione ottenuto è costituito quindi da ben 60 immagini distinte.

5.5.3 Obiettivi iniziali

Vengono di seguito presentati gli obiettivi iniziali della validazione indicando e spiegando il valore assegnato a ciascun criterio:

- **Coerenza tematica:** 4.0 su 5, indica che la maggior parte delle immagini dovrebbe essere chiaramente riconducibile ai *prompt* forniti.
- **Assenza di elementi indesiderati:** 4.5 su 5, indica che la *pipeline* è efficace nel rispettare le restrizioni imposte dai *prompt* negativi, riducendo al minimo eventuali inclusioni di elementi o temi non richiesti.
- **Qualità visiva:** 4.0 su 5, indica che le immagini sono visivamente piacevoli, ben definite e prive di artefatti (imperfezioni, anomalie visive) significativi che possano compromettere l'esperienza visiva dell'utente.

5.5.4. Raccolta dei feedback

La raccolta dei *feeback* è avvenuta sottoponendo a ciascun utente le immagini del campione mediante un *form*, attraverso il quale hanno avuto la possibilità di esprimere il proprio giudizio assegnando un punteggio a ciascun criterio stabilito.

Tale raccolta ha fornito all'attività di validazione un insieme di ben 900 punteggi da analizzare.

5.5.5. Calcolo dei risultati

Sulla base dei 300 *feedback* ottenuti, si è provveduto a calcolare la media dei punteggi assegnati dagli utenti per ogni criterio di valutazione.

Per ciascun criterio, il calcolo della media aritmetica è stato effettuato secondo la formula:

$$Media\ criterio\ K = \frac{somma\ dei\ punteggi\ del\ criterio\ K}{numero\ immagini \times numero\ di\ utenti}$$

L’applicazione di tale formula ai risultati ottenuti per ciascun criterio ha fornito i seguenti punteggi complessivi:

- **Coerenza tematica**: 4.4 ± 0.6
- **Assenza di elementi indesiderati**: 4.6 ± 0.5
- **Qualità visiva**: 4.1 ± 0.6

5.5.6. Analisi dei risultati

I risultati sono stati confrontati con gli obiettivi iniziali presentati e si può concludere che i risultati ottenuti sono coerenti con essi: la *pipeline* generativa ha mostrato un'ottima aderenza ai *prompt* negativi, una buona coerenza tematica e le immagini generate sono prive di anomalie visive tali da compromettere l'esperienza visiva dell'utente.

5.5.7. Conclusioni

La validazione qualitativa ha dimostrato che la *pipeline* generativa sviluppata con *ComfyUI* è in grado di produrre immagini tematicamente coerenti prive di elementi indesiderati e con una qualità visiva generalmente buona.

Le aree di miglioramento identificate includono la riduzione degli artefatti (imperfezioni, anomalie visive) per aumentare ulteriormente la qualità estetica delle immagini generate.

5.5.8. Ulteriori considerazioni

Coinvolgendo esperti del settore e utilizzando criteri di valutazione ancor più dettagliati (qualità delle proporzioni, qualità dei particolari…), è possibile ottenere una valutazione approfondita e affidabile delle prestazioni della *pipeline*.

I *feedback* raccolti attraverso il processo di validazione possono fornire indicazioni preziose per ulteriori ottimizzazioni e miglioramenti, contribuendo a sviluppare una soluzione generativa robusta e di alta qualità.

Conclusione

La presente tesi ha esplorato la creazione e la validazione di una *pipeline* generativa, realizzata con il software *ComfyUI*, finalizzata alla generazione automatizzata di immagini nel settore della moda.

Nei passaggi che coinvolgono lo sviluppo di una *pipeline* di questo tipo, occupa un ruolo importante il processo di addestramento dei modelli generativi, mediante *dataset* di immagini elaborate, affinché questi possano ampliare le loro capacità ed aumentare fedeltà e coerenza stilistica.

Sviluppare tali *dataset* è un processo laborioso che si è riusciti ad automatizzare attraverso l'implementazione di un applicativo software, sviluppato in Python, che elabora le immagini e le etichetta processandole con un modello di IA dedicato.

Attraverso l'implementazione di un applicativo software, anch'esso sviluppato in Python, si è riusciti a fornire un'interfaccia intuitiva e funzionale che consenta agli utenti di interagire efficacemente con la *pipeline* sviluppata.

L'analisi dei requisiti funzionali e non funzionali di tali applicativi ha permesso di costruire dei sistemi robusti e flessibili, capaci di gestire le richieste dell'utente, di visualizzare i risultati ottenuti e scaricali secondo formati opportuni.

È stata effettuata una validazione qualitativa della *pipeline* generativa, condotta tramite *feedback* forniti da un insieme di utenti con competenze specifiche nel settore della moda che hanno valutato un campione rappresentativo di immagini generate dalla *pipeline* presentata.

L'analisi dei risultati ha evidenziato la capacità del sistema di produrre immagini coerenti con i temi specificati e prive di elementi indesiderati. Nonostante alcuni artefatti visivi rilevati, la qualità generale delle immagini è risultata soddisfacente, dimostrando l'ottimo potenziale della *pipeline* generativa sviluppata.

L'adozione di un approccio basato su modelli di Intelligenza Artificiale, come i modelli di Diffusione e i modelli *LoRA*, ha permesso, quindi, di sviluppare uno strumento dotato di ottime capacità generative.

La qualità e la coerenza stilistica mostrata evidenziano, inoltre, il peso innovativo che una *pipeline* di questo tipo può apportare nel processo creativo di contenuti multimediali.

In sintesi, il lavoro svolto rappresenta un significativo passo avanti nell'integrazione di tecnologie di Intelligenza Artificiale nel settore della moda, aprendo la strada a future applicazioni e sviluppi. Risulta chiaro che strumenti generativi di questo tipo possono trovare spazio in tantissimi ambiti e settori industriali. Pertanto, la continua evoluzione delle tecnologie di IA e la loro integrazione nei processi creativi permetteranno di rivoluzionare il modo in cui concepiamo e realizziamo contenuti visivi, rendendo possibili nuovi livelli di innovazione e creatività.

Indice delle figure

Bibliografia e sitografia

Parlamento europeo. *Che cos'è l'intelligenza artificiale?*. 2023. https://www.europarl.europa.eu/topics/it/article/20200827STO85804/che-cos-e-l-intelligenza-artificiale-e-come-viene-usata

Red Hat. *Cos'è l'IA generativa?*. 2024. https://www.redhat.com/it/topics/ai/what-is-generative-ai

Amazon Web Services. *Cos'è Stable Diffusion?*. 2024. https://aws.amazon.com/it/what-is/stable-diffusion/

Amazon Web Services. *Cos'è il Machine Learning?*. 2024. https://aws.amazon.com/it/what-is/machine-learning/

StabilityAI. *SD-XL 1.0-base Model Card*. 2023. https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0

CivitAI, denrakeiw. *Adidas (Tracksuit) [SDXL]*. 2023. https://civitai.com/models/209918/adidas-tracksuit-sdxl

CivitAI, Clara1. *Parkasite-XL*. 2024. https://civitai.com/models/382600/parkasite-xl

Punya, Steve Norman. *[Guide] Train Custom Lora for Fashion Clothes in Stable Diffusion*. 2024. https://punya.ai/blog/post/custom-lora-stable-diffusion-fashion-clothes-commercial

StabilityAI. *stabilityai/sdxl-vae*. 2023. https://huggingface.co/stabilityai/sdxl-vae

Samael1976, CivitAI. *4x NMKD Superscale*. 2023. https://civitai.com/models/141491/4x-nmkd-superscale

Gradio. *Gradio*. 2024. https://www.gradio.app/

OpenCV. *OpenCV*. 2024. https://opencv.org/

Salesforce. *Salesforce/blip-image-captioning-base*. 2023. https://huggingface.co/Salesforce/blip-image-captioning-base

HuggingFace. *Transformers*. 2024. https://huggingface.co/docs/transformers/

Ringraziamenti

È doveroso dedicare questo spazio del mio elaborato alle persone che hanno contribuito, con il loro supporto, alla realizzazione dello stesso.

In primo luogo, ringrazio il mio relatore Porfirio Tramontana, per la sua disponibilità, per i suoi indispensabili consigli e per l'attenzione mostrata durante tutto il percorso di tirocinio e di stesura dell'elaborato.

Ringrazio l'azienda Kineton srl e in particolar modo il mio tutor aziendale Paolo Nappi per l'opportunità che mi è stata concessa, per l'attenzione e la disponibilità mostrata durante l'esperienza di tirocinio di cui è frutto questo elaborato.

Ringrazio infinitamente la mia famiglia e i miei amici che hanno sempre creduto in me e mi hanno sempre sostenuto. È grazie a voi che ho superato i momenti più difficili e tagliato i traguardi più importanti.

Un grazie di cuore ai miei colleghi, con cui ho condiviso l'intero percorso universitario. Siete stati veri e propri fratelli. Grazie a voi porterò con me ricordi bellissimi.

Infine, dedico questa tesi a me stesso, ai miei sacrifici e alla mia tenacia che mi hanno permesso di arrivare fin qui. Che possa essere l'inizio di una lunga e brillante carriera.