POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria del Cinema e dei Mezzi di Comunicazione

Tesi di Laurea Magistrale

IA Generativa nel Cinema: caso studio sulla serie “Il partigiano Libero”

Relatrice

Prof.ssa Tatiana MAZALI

Candidato

Giuseppe BRUNO

Tutor aziendale

Motion Pixel

Stefano SBURLATI

Aprile 2025

Abstract

Negli ultimi anni, l'Intelligenza Artificiale Generativa ha pervaso ogni settore dell'industria tecnologica, incluso quello audiovisivo, che ne sta esplorando le potenzialità e i vantaggi.

A partire dalla volontà di ANPI Chiomonte di raccontare le vicende di un partigiano che ha partecipato alla liberazione d'Italia e dall'interesse di Motion Pixel nell'esplorare le potenzialità dell'IA Generativa per la creazione di video, questo lavoro di tesi analizza e studia l'applicazione di tali tecnologie nella produzione di una serie video sperimentale. L'obiettivo principale è valutare le potenzialità e i limiti dell'IA Generativa nella generazione di immagini e video, applicandone le diverse tecniche all'intero processo produttivo.

La ricerca prevede lo studio di tecnologie open-source come Stability Matrix, ComfyUI, Stable Diffusion e CogVideo, con particolare attenzione all'impiego di modelli LoRA per migliorare la consistenza stilistica. L'aspetto principale del lavoro è la definizione di una pipeline di produzione efficiente e ottimizzata per la generazione di contenuti a partire da semplici descrizioni testuali, che includa l'utilizzo di modelli personalizzati per mantenere la coerenza stilistica e la fedeltà visiva dei personaggi.

Il risultato di questa analisi è una riflessione sul presente e sul futuro dell'IA Generativa nell'industria cinematografica e sulle opportunità offerte per l'innovazione nello storytelling audiovisivo.

Keywords Intelligenza Artificiale, Generative AI, Cinema, Stable Diffusion, ComfyUI, Stability Matrix, CogVideo, LoRA

Abstract in inglese

In recent years, Generative Artificial Intelligence has permeated every sector of the technology industry, including the audiovisual one, which is exploring its potential and advantages.

Based on ANPI Chiomonte's desire to recount the events of a partisan who took part in Italy's liberation, and Motion Pixel's interest in exploring the potential of Generative AI for video creation, this thesis analyzes and examines the application of such technologies in the production of an experimental short film. The main objective is to assess the capabilities and limitations of Generative AI in image and video generation, applying various techniques throughout the entire production process.

The research focuses on the study of open-source technologies such as Stability Matrix, ComfyUI, Stable Diffusion, and CogVideo, with particular attention to the use of LoRA models to enhance stylistic consistency. A key aspect of this work is the definition of an efficient and optimized production pipeline for generating content from simple textual descriptions, integrating customized models to maintain stylistic coherence and visual fidelity of the characters.

The result of this analysis is a reflection on the present and future of Generative AI in the film industry and the opportunities it offers for innovation in audiovisual storytelling.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Con il termine *Intelligenza Artificiale Generativa* (dall'inglese *Generative Artificial Intelligence* o *Generative AI*) ci si riferisce a una categoria di sistemi di Intelligenza Artificiale capaci di generare contenuti nuovi e originali attraverso l'uso di modelli di apprendimento profondo. Questo sottoinsieme delle tecnologie di intelligenza artificiale si distingue per la sua possibilità di generare output come testi, immagini, musica, video e altro, a partire da semplici input forniti dall'utente. L'importanza dello studio delle applicazioni di IA generativa risiede nella loro capacità di rivoluzionare numerosi settori, trasformando le metodologie di lavoro finora utilizzate e offrendo nuove opportunità per l'innovazione. Ad esempio, le applicazioni di IA Generativa possono automatizzare la creazione di contenuti, migliorare i processi decisionali e generare esperienze utente personalizzate [1].

L'obiettivo di questo lavoro di tesi è quello di affrontare le fasi di creazione di un prodotto audiovisivo, generato utilizzando strumenti di Intelligenza Artificiale Generativa, per comprenderne le potenzialità e studiarne le possibili applicazioni pratiche, nell'ambito di una produzione cinematografica. Si analizzerà come l'IA generativa possa influenzare le dinamiche del processo creativo cinematografico, esplorando le tecniche utilizzate e le implicazioni estetiche e narrative derivanti da questa tecnologia innovativa. Inoltre, si prenderà in considerazione come l'adozione di tali strumenti possa portare a una nuova era di storytelling visivo.

1.1 Diversi tipi di IA Generativa

Le tre principali tipologie di Intelligenza Artificiale Generativa studiate e analizzate nel processo di produzione di questo lavoro di tesi sono quelle per la generazione di immagini a partire da input di tipo testuale e per la generazione di video a partire da input contenenti testo o immagini. Se ne descrive il funzionamento di seguito.

1.1.1 Text-To-Image

Un'IA generativa di tipo Text-To-Image è un modello di intelligenza artificiale progettato per creare immagini a partire da descrizioni testuali. Questi modelli utilizzano tecniche avanzate di deep learning ed elaborazione del linguaggio naturale (detti NLP, *Natural Language Processing*) per interpretare il testo fornito, comprenderne il significato, gli elementi chiave e le relazioni tra di essi. Successivamente, il modello converte la descrizione in una rappresentazione semantica che rappresenta visivamente i concetti descritti, per poi procedere alla generazione dell'immagine utilizzando reti neurali, come i modelli basati su trasformatori, reti generative avversarie (dette GAN, *Generative Adversarial Networks*) o altri approcci [2].

1.1.2 Text-To-Video

Rispetto a quanto detto precedentemente, va da sé che un'IA generativa di tipo Text-To-Video riesca a generare una sequenza video a partire da una descrizione testuale. A differenza dei generatori di immagini, i modelli Text-To-Video devono generare non solo un'immagine statica, ma sequenze di immagini che si susseguono nel tempo. Ciò implica che il modello deve considerare la coerenza temporale e garantire una transizione fluida tra le varie scene e azioni rappresentate. Attraverso un processo che combina la generazione di fotogrammi statici e la loro sequenzializzazione, il modello produce video coerenti e dinamici. Le architetture comunemente utilizzate includono modelli basati su diffusione e reti neurali ricorrenti, che permettono di gestire il movimento e la continuità temporale nel video [2].

Nel momento in cui questa tesi viene redatta, la limitazione principale dei modelli di generazione video è quella di non riuscire a mantenere sempre la coerenza visiva e l'alta qualità su una sequenza di fotogrammi. Alcune tecnologie innovative come *Sora*, sviluppata da OpenAI, potrebbero superare questi limiti, ma non sono ancora state rese disponibili al pubblico [3].

1.1.3 Image-To-Video

Ciò che momentaneamente supera i limiti dei modelli Text-To-Video, descritti precedentemente, e che ha rappresentato uno snodo cruciale per la realizzazione di questo lavoro di tesi, sono i modelli di IA generativa di tipo Image-To-Video. Questi modelli si basano su un'immagine iniziale come condizione per generare una sequenza video che mantenga coerenza stilistica e semantica con l'input. A differenza dei modelli Text-To-Video, che devono costruire interamente il contenuto visivo a partire da un testo, i modelli Image-To-Video sfruttano l'informazione visiva preesistente per generare il movimento e l'evoluzione della scena.

L'efficacia di questo approccio risiede nell'integrazione di reti neurali convoluzionali con architetture basate su trasformatori. L'utilizzo di autoencoder variazionali 3D (VAE, *Variational Autoencoder*), che comprimono i video lungo le dimensioni spaziali e temporali, migliora la fedeltà del video ed evita i problemi di coerenza e continuità che si avevano coi modelli precedenti. Inoltre, poiché i modelli Text-To-Video spesso non hanno descrizioni testuali dettagliate, i modelli Image-To-Video possono incorporare descrizioni che migliorano la comprensione semantica e visiva, permettendo una generazione video più accurata. Ciò è dovuto all'integrazione di dati testuali provenienti da video esistenti: i modelli di tipo Image-To-Video possono sfruttare video già esistenti e ben descritti per generare nuovi video, utilizzando le loro informazioni visive per garantire una maggiore qualità e coerenza [4].

1.2 Stato dell'arte

Nel parlare di stato dell'arte delle tecnologie di Intelligenza Artificiale Generativa, è necessario fare un appunto su quello che riguarda la rapidità dell'evoluzione delle tecniche, dei modelli e dei paradigmi di generazione. Quello che viene scritto in questo lavoro potrebbe divenire obsoleto nel giro di pochi mesi; nel tempo che intercorre tra la stesura di questa tesi e il momento della sua presentazione, l'annuncio di una nuova tecnologia di IA Generativa o di un nuovo metodo di generazione potrebbe stravolgere il mercato e cambiarne completamente i paradigmi di utilizzo. Le informazioni contenute in questo lavoro di tesi sono aggiornate al mese di Febbraio 2025.

Generazione di immagini

Ad oggi, il panorama della generazione di immagini è caratterizzato per la maggior parte da modelli basati su tecniche di diffusione, che riescono a produrre risultati di alta qualità con un elevato controllo dello stile. L'integrazione di meccanismi di *prompt engineering* avanzato e adattamenti specifici, come l'utilizzo di modelli LoRA, ha portato a una personalizzazione di stile, dettaglio e composizione in maniera estremamente precisa [5].

DALL·E è un modello di IA Generativa, sviluppato da OpenAI, che utilizza il modello Text-To-Image per la generazione delle immagini. Dalla versione 3 è stato integrato all'interno di ChatGPT, il noto assistente testuale della stessa azienda basato sul modello GPT-4. Grazie a questa integrazione, è possibile fornire la descrizione in un linguaggio naturale, che verrà poi sintetizzata in *prompt* direttamente da GPT, rendendo più fedele e accurata la generazione anche per le richieste vaghe o formulate in modo non troppo accurato. Come descritto precedentemente, e ponendosi come strumento di generazione di immagini all'avanguardia, DALL·E 3 utilizza un'architettura basata su modelli di diffusione, che consente di migliorare la qualità rispetto alle reti generative avversarie (GAN). Con questa tecnica, l'immagine iniziale è composta da rumore casuale e viene progressivamente affinata fino a ottenere una rappresentazione chiara e dettagliata, consentendo di ottenere immagini con minori artefatti visivi e una migliore coerenza semantica.

DALL·E è accessibile attraverso i piani a pagamento di ChatGPT, il suo utilizzo è subordinato alle linee guida e alle *policy* di OpenAI e, più in generale, è un modello chiuso di cui il codice sorgente e i pesi del modello non sono disponibili pubblicamente [6].

Midjourney è l'alternativa più conosciuta a DALL·E e ChatGPT. Sviluppata dall'omonima azienda, la sua prima versione è stata rilasciata nel 2022 ed è diventato popolare grazie al *chat bot* presente su un server Discord, che gli utenti possono messaggiare per generare immagini, chiedere assistenza e discutere con altri membri della community. Sin dalla sua nascita, Midjourney è disponibile in beta pubblica. L'ultima versione rilasciata è la 6.1, pubblicata il 31 luglio 2024, la cui maggiore novità è stata l'introduzione di una nuova interfaccia Web dedicata, che implementa tutte le funzionalità già presenti su Discord (la cui interfaccia continua ad essere presente) e mantiene sincronizzati ambidue gli ambienti. Ciò che distingue principalmente Midjourney dal rivale DALL·E è la sua forte impronta artistica e stilistica, che lo rende maggiormente apprezzato da designer, illustratori e lavoratori delle industrie creative digitali. La sua estetica

si pone come più raffinata e dettagliata, con la capacità di produrre immagini di forte impatto visivo anche a partire da prompt testuali molto semplici.

Attualmente, anche Midjourney è un modello chiuso a pagamento, il cui codice sorgente e i pesi del modello non sono disponibili pubblicamente. Così come DALL·E, il suo utilizzo è regolato dalle *policy* di utilizzo, che non consentono la generazione di contenuti sensibili [7].

Generazione di video

Parallelamente, negli scorsi due anni anche la generazione di video ha compiuto passi enormi, riuscendo quasi ad arrivare allo stesso livello qualitativo attualmente raggiunto per la generazione di immagini. I modelli Text-To-Video riescono a riprodurre video di lunga durata ad alta risoluzione, i quali presentano una maggior coerenza narrativa tra i vari frame [8].

Runway è attualmente la piattaforma di IA Generativa più utilizzata e più conosciuta nel mercato delle tecnologie di generazione. Sebbene l'azienda sia stata fondata nel 2018, la prima versione del modello generativo, chiamata Gen-1, è stata rilasciata nel 2023, con cui sono stati introdotti i modelli a diffusione latente [9]. L'ultima versione disponibile, la Gen-3 Alpha, è stata rilasciata nel mese di Giugno 2024 e include numerosi miglioramenti dal punto di vista della fedeltà visiva, della consistenza delle immagini e dell'accuratezza dei movimenti, oltre ad essere stata allenata su nuove infrastrutture realizzate appositamente per l'addestramento modale su larga scala. Runway è ad oggi uno dei modelli di IA Generativa più completi presenti sul mercato: utilizza infatti diverse tecniche per mantenere sia la fedeltà del contenuto che quella della struttura durante la generazione dei video. Queste comprendono:

- **Stime di profondità monoculare** che consentono di rappresentare la struttura del video attraverso caratteristiche geometriche e dinamiche della scena, per mantenere la posizione spaziale degli oggetti.
- **Modello generativo condizionale** che permette di semplificare il video in una funzione probabilistica del tipo $p(x \mid s, c)$, dove x è il video, s la rappresentazione della struttura e c la rappresentazione del contenuto; questo metodo assicura che il video rispetti le condizioni date in input per quanto riguarda la strutta e il contenuto, attraverso un controllo dettagliato sulle caratteristiche del risultato finale.
- ***Guidance method* personalizzato**, ispirato ai *guidance method* senza classificatori (ovvero che utilizzano una combinazione di previsioni basate

su input specifici e previsioni incondizionate); consente agli utenti di modificare il comportamento del modello, decidendo quanto quest'ultimo possa generare contenuti originali scostandosi dalla descrizione iniziale, o quanto debba essere fedele alle caratteristiche fornite nel prompt.

Sebbene anche Runway sia un sistema chiuso, il cui codice sorgente è proprietario e non disponibile pubblicamente, l'azienda pone il proprio modello di business su un'offerta *Software as a Service* (SaaS), proponendo diversi livelli di abbonamento con accesso a funzionalità diverse. Il piano base è gratuito e offre accesso al modello di generazione con un limite massimo di circa 25 secondi, permettendo praticamente solo di esplorare le funzionalità della piattaforma [10].

Sora è il modello di generazione video di OpenAI, la stessa azienda di ChatGPT e DALL·E. Annunciata ufficialmente nel 2024 e non ancora disponibile pubblicamente, Sora integra gli aspetti caratterizzanti degli altri due modelli, con tutti i pregi e le migliorie date dall'ampio utilizzo delle stesse. Questo modello consente di generare video in alta definizione utilizzando input testuali, visuali e video esistenti, con una durata massima di 20 secondi per ogni video. Così come gli altri modelli di ultima generazione citati, anche Sora è un modello a diffusione, il cui funzionamento prevede che i video siano inizialmente composti da rumore statico e si trasformino gradualmente nel risultato finale [8]. Le tecniche implementate per garantire risultati di alta qualità includono:

- **Tecniche di recaptioning**, ispirate a DALL·E 3, che consentono al modello di interpretare accuratamente le descrizioni testuali fornite dagli utenti nella generazione dei video.

- **Rappresentazione a *patches* visivi**, in cui le patches sono segmenti di video che vengono compressi in modo simile a quanto accade con i *token* nei modelli LLM; questa tecnica consente di avere una rappresentazione più compatta dei dati visivi, per poter mantenere un livello elevato di dettaglio durante la generazione, potendo analizzare ed elaborare ogni patch come un video a sé stante [11].

1.2.1 Tecnologie di IA Generativa open source

Nelle fasi iniziali della lavorazione di questo progetto di tesi, ci si è interrogati anche in merito alla possibilità di utilizzare tecnologie di Intelligenza Artificiale Generativa open source e gratuite. Gli strumenti open source sono liberamente

accessibili a chiunque, con la possibilità di scaricare, modificare e distribuire il codice senza restrizioni, oltre ad avere costi diretti[1] azzerati per il loro utilizzo.

Visto il fine di ricerca e sperimentazione, la scelta di strumenti open source avrebbe garantito il perseguimento degli obiettivi di libertà delle informazioni e di condivisione dei materiali con la comunità. Ricerca scientifica e ideologia open source sono due argomenti strettamente correlati tra loro. La natura aperta degli strumenti consente a diversi ricercatori e sviluppatori di contribuire con miglioramenti e nuove funzionalità, permettendo così di avere sviluppi più rapidi e accelerando l'avanzamento della ricerca scientifica. Dall'altro lato, e soprattutto per quanto riguarda la ricerca in campo informatico, la varietà di esperienze e prospettive che gli utilizzatori apportano aiuta a trovare soluzioni condivise a problemi complessi che un singolo ricercatore potrebbe non essere in grado di affrontare, creando grandi reti di ricercatori che incentivano la cooperazione internazionale e multidisciplinare, contribuendo a creare una cultura di condivisione e a lasciare un forte impatto sulla comunità.

Un altro aspetto fondamentale per questo lavoro di tesi, garantito dalle tecnologie open source, è stato quello di poter accedere liberamente ai modelli, senza avere controllo o restrizioni su contenuti sensibili o controversi. Nella scelta delle tecnologie, come verrà descritto più dettagliatamente in seguito, l'utilizzo di strumenti open source ha garantito massima libertà sulla rappresentazione di scene e vicende ambientate nel contesto storico della Seconda guerra mondiale. Come verrà approfondito in seguito, la scelta di strumenti open source non solo ha garantito la massima flessibilità nella produzione audiovisiva, ma si inserisce anche in un più ampio dibattito sulle opportunità e i rischi di questi modelli [12].

1.3 Etiche di utilizzo delle tecnologie di IA Generativa

L'utilizzo delle tecnologie di Intelligenza Artificiale Generativa solleva questioni etiche rilevanti. La maggior parte dei modelli riflettono il contesto culturale dell'ambiente in cui sono stati sviluppati: i *dataset* utilizzati per l'addestramento dei modelli di Intelligenza Artificiale sono non privi di *bias* e pregiudizi storici

[1] Si specificano i costi "diretti" nel senso che in questo caso si aggiungono i costi indiretti relativi al dispendioso consumo elettrico che la generazione di contenuti multimediali comporta, argomento che questo lavoro di tesi esula di trattare.

e culturali, dettati da stereotipi ed esclusioni che chi si occupa di creare quel *dataset* inserisce, più o meno volontariamente.

Se, in teoria, la responsabilità per la diversità è condivisa tra vari attori e livelli della società, quindi nel caso delle tecnologie di IA tra fornitori, utilizzatori e ricercatori, nella pratica ciò avviene difficilmente. Solitamente, i fornitori di strumenti di IA tendono a delegare agli utenti finali il compito e la responsabilità di garantire la diversità nei contenuti e promuovere un utilizzo degli strumenti di IA senza pregiudizi. Per il momento, i fornitori si limitano a divulgare le fonti dei loro dati di addestramento e le limitazioni dei loro sistemi, consentendo agli utenti di comprendere e contestare eventuali *bias*, ma senza intervenire direttamente all'eliminazione degli stessi. L'uso improprio dell'IA in contesti creativi e audiovisivi non solo pone interrogativi sull'autenticità delle opere, sull'autorialità dei contenuti e sul diritto d'autore, ma esiste anche il pericolo che l'utilizzo indiscriminato di strumenti di Intelligenza Artificiale Generativa possa standardizzare l'estetica e il linguaggio visivo, riducendo la diversità artistica anziché arricchirla [13].

Un altro aspetto importante che si è presentato nel corso di questo lavoro riguarda l'utilizzo di modelli per la creazione di personaggi fittizi generati con l'IA Generativa. Per la creazione di questi modelli, sono stati utilizzati *dataset* personalizzati contenenti un'unione di immagini di persone reali e immagini di persone generate da modelli preesistenti. L'addestramento di modelli LoRA (*Low-Rank Adaptation*), di cui si discuterà in seguito, creati a partire da persone reali comporta problemi relativi alla privacy e alla proprietà intellettuale dell'immagine. Se da un lato ciò consente di creare figure fittizie e garantire una maggior coerenza del racconto e dell'apparato stilistico, dall'altro porta pericolosamente alla ridefinizione dell'identità visiva di individui senza il loro consenso.

Nel corso dello sviluppo di questo progetto, ci si è ritrovati spesso ad affrontare delle scelte di carattere etico volte a mitigare questi rischi, adottando un approccio quanto più consapevole nella selezione dei *dataset* e nella generazione dei personaggi. L'approccio adottato ha cercato di bilanciare le potenzialità offerte dalle tecnologie di IA Generativa con una riflessione critica sui loro limiti e sulle implicazioni etiche del loro utilizzo. Queste verranno descritte più approfonditamente nei capitoli successivi.

1.4 Struttura della tesi

Questa tesi si struttura in cinque capitoli, descritti come segue:

- **Capitolo 2. Contesto del progetto**: vengono descritti gli attori principali coinvolti nell'ideazione e produzione del prodotto: l'azienda Motion Pixel Srl, l'associazione ANPI Sezione Chiomonte Alta Valle Susa; sono inoltre descritte le informazioni sugli obiettivi del lavoro di tesi.
- **Capitolo 3. Tecnologie open source utilizzate**: sono descritti nel dettaglio tutti gli strumenti di IA Generativa utilizzati, con particolare attenzione a quelli di tipo open source.
- **Capitolo 4. Produzione**: è il nucleo vero e proprio di questo lavoro di tesi; in seguito al lavoro di ricerca e analisi degli strumenti, in questo capitolo viene illustrata passo per passo la *pipeline* di produzione seguita per la creazione del cortometraggio.
- **Capitolo 5. Conclusioni**: vengono discussi i risultati del progetto, i limiti dei modelli open source utilizzati e vengono date delle ipotesi di utilizzo degli strumenti di IA Generativa nel cinema.

Capitolo 2

Contesto del progetto

2.1 Realtà coinvolte

2.1.1 Motion Pixel

L'azienda **Motion Pixel Srl** è una piccola e dinamica attività attiva da oltre 15 anni, fondata da Stefano Sburlati, che lavora nella produzione di video e contenuti digitali per aziende, enti pubblici e cooperative. Nel corso degli anni l'azienda si è specializzata nella realizzazione di contenuti immersivi a 180° e 360° [14]. L'ultima produzione dello studio è il cortometraggio immersivo *Sweet end of the World!*, diretto da Stefano Conca Bonizzoni, vincitore del premio "Rai Cinema Channel VR" al FilmMaker Festival [15].

Figura 2.1: Logo di Motion Pixel

Con l'avvento delle tecnologie di Intelligenza Artificiale Generativa, l'azienda ha da subito deciso di incentrare la propria ricerca e sviluppo sullo studio di questa tecnologia, cogliendone e apprendendo tutte le innovazioni del settore.

2.1.2 ANPI Chiomonte

L'**Associazione Nazionale Partigiani Italiani - Sezione Chiomonte Alta Valle Susa** nasce nel 1970 per volere dei partigiani che hanno combattuto in Alta Valle di Susa nel corso della Seconda guerra mondiale. Sin dalla sua fondazione, la sezione si è impegnata nel preservare la memoria della Resistenza, attraverso l'organizzazione di iniziative di carattere culturale, storico e didattico, e con l'obiettivo di tramandare i valori di antifascismo, libertà e democrazia [16].

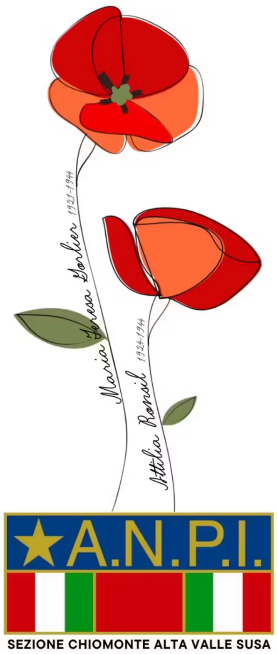


Figura 2.2: Logo di ANPI Chiomonte Alta Valle Susa

Negli ultimi anni, ANPI Chiomonte ha cercato di coinvolgere le nuove generazioni nel percorso di trasmissione della memoria. Da questa necessità, è nato il desiderio di realizzare un prodotto audiovisivo che presenti caratteri originali e moderni, da condividere successivamente attraverso i canali *social* di recente creazione dell'associazione stessa.

2.2 Concept e obiettivi del cortometraggio

In seguito alle premesse fatte poc'anzi, nei primi mesi del 2024 è nato un dialogo tra l'azienda Motion Pixel e l'associazione ANPI Chiomonte volto alla realizzazione di un prodotto audiovisivo che rendesse omaggio al comandante Cesare Alvazzi del Frate, zio del fondatore dell'azienda Stefano Sburlati e protagonista della Resistenza nell'Alta Valle di Susa della più importante battaglia, combattuta il 17 luglio del 1944 tra il Monte Triplex e il Col Basset [17].

ANPI Chiomonte ha fornito le proprie conoscenze storiche e narrative, occupandosi della scrittura del concept del cortometraggio e della sceneggiatura degli episodi, per una miniserie che inizialmente prevedeva la realizzazione di 20 episodi. Invece, Motion Pixel ha messo in campo i propri studi svolti in materia di IA Generativa, oltre a fornire la propria infrastruttura tecnologica.

L'inizio di questo lavoro di tesi coincide con l'inizio della preproduzione della serie. Il candidato ha cominciato il proprio percorso di tesi all'interno dell'azienda durante il mese di Aprile 2024 e ha partecipato alla produzione del lavoro fino al mese di Gennaio 2025.

L'idea dietro la realizzazione della serie consiste nel ripercorrere il cammino di Cesare Alvazzi Del Frate, raccontando il coraggio e il sacrificio che hanno animato la lotta per la liberazione d'Italia, intrecciando la sua storia con i fatti che scuotevano l'Italia e, più in dettaglio, la Resistenza in alta Valle di Susa. Inizialmente, la serie prevedeva la realizzazione di 20 episodi della durata compresa tra uno e due minuti, datati a partire dal mese di Maggio 1943 fino ad arrivare a pochi giorni dopo la giornata della liberazione d'Italia, più precisamente fino al 27 Aprile 1945. Si riporta di seguito il concept della serie, concordato tra Motion Pixel e ANPI Chiomonte:

Il Comandante Cesarino

Cesare, un giovane studente torinese, sfollato con la famiglia da Torino ad Oulx per sfuggire ai bombardamenti, si avvicina alla resistenza fino a diventarne parte integrante. Attratto dai valori di libertà e giustizia, si unisce alla Resistenza nelle Valli di Susa e Chisone. La sua crescita come comandante partigiano è scandita da battaglie eroiche, relazioni profonde e sfide contro l'oppressione nazifascista.

La serie è pensata per essere realizzata interamente con tecniche di Intelligenza Artificiale Generativa: tutto il video è realizzato con questa tecnica, anche

utilizzando immagini e materiale d'archivio come base di partenza per aumentare l'aderenza storica. Inoltre, per limitare al massimo i dialoghi si è scelto di utilizzare un voice-over che accompagna e racconta il video.

L'idea e la sceneggiatura iniziale sono di Renato Sibille e Roberto Micali, con il prezioso contributo della figlia del partigiano Cesare, Anna Alvazzi Del Frate, che ha messo a disposizione i propri ricordi riguardo ai racconti della vita del padre. Gli autori si sono ispirati con grande aderenza storica alla biografia del protagonista e ai fatti avvenuti: alcune di queste prevedevano scene di vita reale del protagonista, come un episodio riguardante uno sgambetto dato da Cesare a Paolo Gobetti durante un'adunata fascista, o alcune vicende personali del partigiano accadute durante la battaglia del Triplex. Tuttavia, per esigenze narrative, si sono concessi la libertà di evidenziare o, al contrario, tralasciare alcuni aspetti; i personaggi principali sono reali, mentre è possibile che all'interno della serie alcuni dei personaggi secondari siano non reali, mantenendo comunque caratteristiche di verosimiglianza a persone realmente esistite.

A carattere di esempio, si riporta la sinossi del primo episodio della serie, fornito da ANPI Chiomonte e scritto da Renato Sibille, Roberto Micali e Anna Alvazzi Del Frate.

EPISODIO 1.

Maggio 1943. Torino

Mentre Cesare partecipa come avanguardista a un'adunata paramilitare della GIL, fa uno scherzo a Paolo Gobetti che era vicino a lui: mentre era sull'attenti, piega il ginocchio e dà un colpo dietro a quello di Paolo che finisce a terra con una distorsione. Cesare, dispiaciuto, va a trovarlo a casa e conosce sua mamma Ada, la vedova di Piero Gobetti, e scopre un mondo di cultura antifascista.

Successivamente, alcuni limiti di produzione e gestione delle risorse hanno portato alla necessità di rivedere alcuni elementi di produzione:

1. La famiglia Alvazzi Del Frate ha iniziato a manifestare dei dubbi in merito all'utilizzo degli strumenti di Intelligenza Artificiale Generativa per rappresentare il padre attraverso la rielaborazione di immagini d'archivio e fotografie personali. Per questo motivo è stato rimosso ogni riferimento diretto al partigiano Del Frate, mantenendo però le caratteristiche di *character design* del personaggio e l'impianto narrativo originale, inclusi i luoghi, gli eventi e il contesto storico-culturale in cui si svolgono le vicende

2. Per venire incontro alle esigenze produttive di Motion Pixel, a causa dell'assenza di finanziamenti economici e per ottimizzare le risorse disponibili, si è scelto di ridurre il numero di episodi a 10, così da rendere il processo di lavoro più agile e garantire tempi di produzione più rapidi.

Dunque, durante il mese di Ottobre 2024 si è arrivati a una soluzione comune che teneva conto di queste premesse e che ha consentito di poter dare avvio alla fase di produzione. Ai fini di questo lavoro di tesi e per effettuare un'analisi preliminare degli strumenti di IA Generativa, si è deciso di realizzare un video *trailer* della serie completa, della durata di 90 secondi circa, in cui vengono presentati il protagonista della storia (che con la nuova sceneggiatura è diventato il partigiano Libero) e diverse ambientazioni e avvenimenti presenti nella versione finale.

La sceneggiatura del trailer realizzato è disponibile nell'appendice A; la relativa *shotlist*, soprannominata in questo caso "promptlist" (perché è una raccolta non tanto degli *shot* da girare ma dei *prompt* utilizzati per la generazione) è invece consultabile nell'appendice B.

Gli obiettivi principali nella realizzazione del cortometraggio sono stati:

- Realizzare un lavoro che rispettasse in maniera accurata il contesto storico e culturale narrato, rendendo omaggio ai partigiani che presero parte alla Resistenza e mantenendo i fatti storicamente accurati.

- Mantenere una coerenza stilistica sia coi personaggi realizzati, sia con gli ambienti rappresentati, avendo cura che essi fossero coerenti con i luoghi reali in cui sono avvenuti i fatti raccontati.

- Evitare le censure sulla realizzazione di contenuti sensibili posta dagli strumenti di IA Generativa attualmente disponibili pubblicamente, soprattutto per la rappresentazione di armi, scene di guerra e di combattimento.

Si discuteranno nei capitoli successivi le modalità attraverso le quali questi obiettivi sono stati perseguiti.

Capitolo 3

Tecnologie open source utilizzate

In questo capitolo vengono descritti gli strumenti di IA Generativa scelti per la produzione della serie. La scelta più importante effettuata è stata di utilizzare per la maggior parte del lavoro tecnologie open source, come descritto già nel paragrafo 1.2.1. Per ognuno di questi sono mostrate le funzionalità più rilevanti ai fini della produzione del lavoro di tesi e i vantaggi rispetto ad altri strumenti più comuni, che hanno infine portato alla scelta di questi. Per alcuni scopi specifici sono stati utilizzati anche strumenti chiusi e a pagamento: anche per questi viene fornita una breve descrizione delle principali funzionalità.

3.1 Piattaforma di gestione: Stability Matrix

Già dai primi momenti di analisi delle varie possibilità di avanzamento dei lavori, è stato chiaro che sarebbe stato necessario utilizzare un numero elevato di strumenti di IA generativa, ciascuno da utilizzare per uno scopo differente dall'altro. Inoltre, l'installazione e la manutenzione di tali strumenti si sono rivelate spesso complesse, con nuovi aggiornamenti rilasciati quasi quotidianamente, dipendenze tra software diversi da mantenere costantemente e diverse configurazioni da gestire.

La soluzione migliore per affrontare queste difficoltà è stata **Stability Matrix**, una piattaforma sviluppata da Lykos AI per centralizzare la gestione delle principali interfacce di Stable Diffusion. Questa piattaforma consente di installare al

suo interno versioni portatili[2] di diverse applicazioni, tra cui ComfyUI (descritto nel paragrafo 3.2) e Koya_ss (paragrafo 3.4), come mostrato nella figura 3.1.

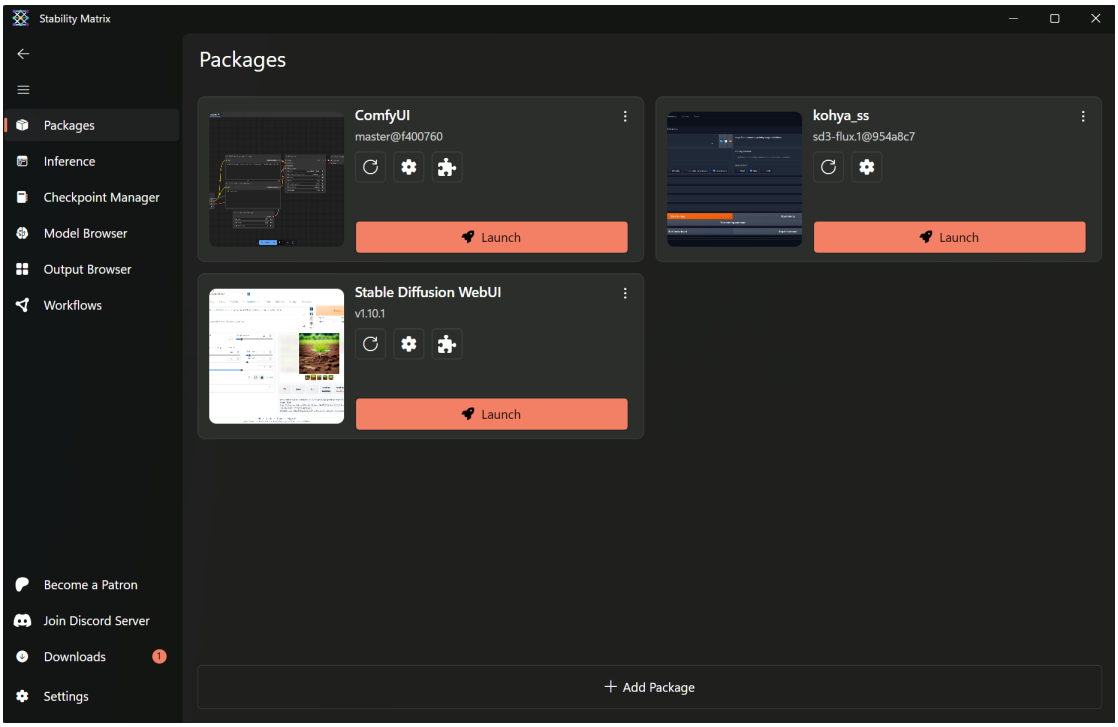


Figura 3.1: Interfaccia principale di Stability Matrix, con in evidenza gli strumenti di IA Generativa installati

È inoltre possibile scaricare direttamente da qui i vari modelli di generazione, grazie all'integrazione con Civitai[3] e Hugging Face[4], da due finestre dedicate (figura 3.2).

Infine, Stability Matrix si occupa anche dell'installazione e della manutenzione di Python e di tutti i pacchetti necessari ai vari strumenti installati, semplificando e rendendo più efficiente l'utilizzo dei vari strumenti (figura 3.3).

[2]Un'applicazione portatile è un *software* applicativo che non necessita di essere installato nel sistema operativo su cui viene eseguito. Programmi di questo tipo possono essere memorizzati in un'unica cartella all'interno del computer e possono essere spostati in altri percorsi senza pregiudicare il loro funzionamento.

[3]Piattaforma per la condivisione di modelli personalizzati di Stable Diffusion, creati e condivisi dalla comunità. URL: https://civitai.com

[4]Archivio per modelli base di machine learning e IA generativa. URL: https://huggingface.co

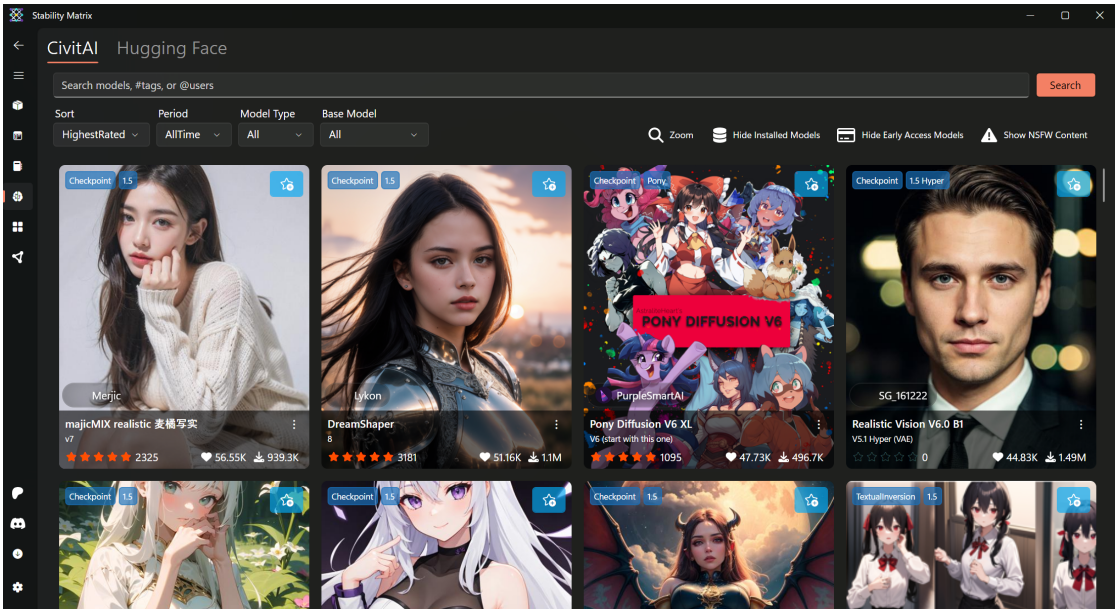


Figura 3.2: Integrazione di Civitai all'interno di Stability Matrix

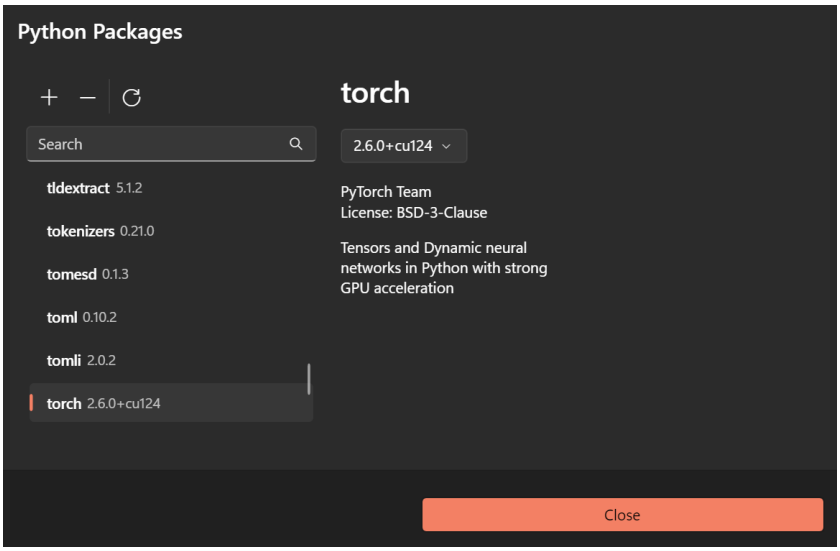


Figura 3.3: Pacchetto torch di Python installato all'interno di uno strumento presente su Stability Matrix

3.2 Interfaccia: ComfyUI

ComfyUI è un'interfaccia grafica per l'utilizzo semplificato degli strumenti di Intelligenza Artificiale Generativa, sviluppata dal collettivo comfyanonymous

e altri contributori all'interno della pagina GitHub. L'interfaccia di ComfyUI costituisce un ambiente per costruire ed eseguire contenuti generativi chiamati *workflow*. Nel contesto di lavoro, un workflow è un insieme di oggetti di programmazione chiamati nodi, connessi tra di loro per formare una rete, come mostrato nella figura 3.4. Un workflow ComfyUI può generare ogni tipo di contenuto: immagini, video, audio, modelli di IA, e così via.

Il concetto alla base del funzionamento di ComfyUI è quello della programmazione a nodi. Fornendo un ambiente di programmazione visiva, il software consente di progettare sistemi complessi senza la necessità di scrivere direttamente codici di programmazione. Questo paradigma si ispira alle più moderne applicazioni di effetti visivi utilizzate nell'industria cinematografica, come i programmi 3D Nuke e Blender, il motore grafico real-time Unreal, e molti altri [18].

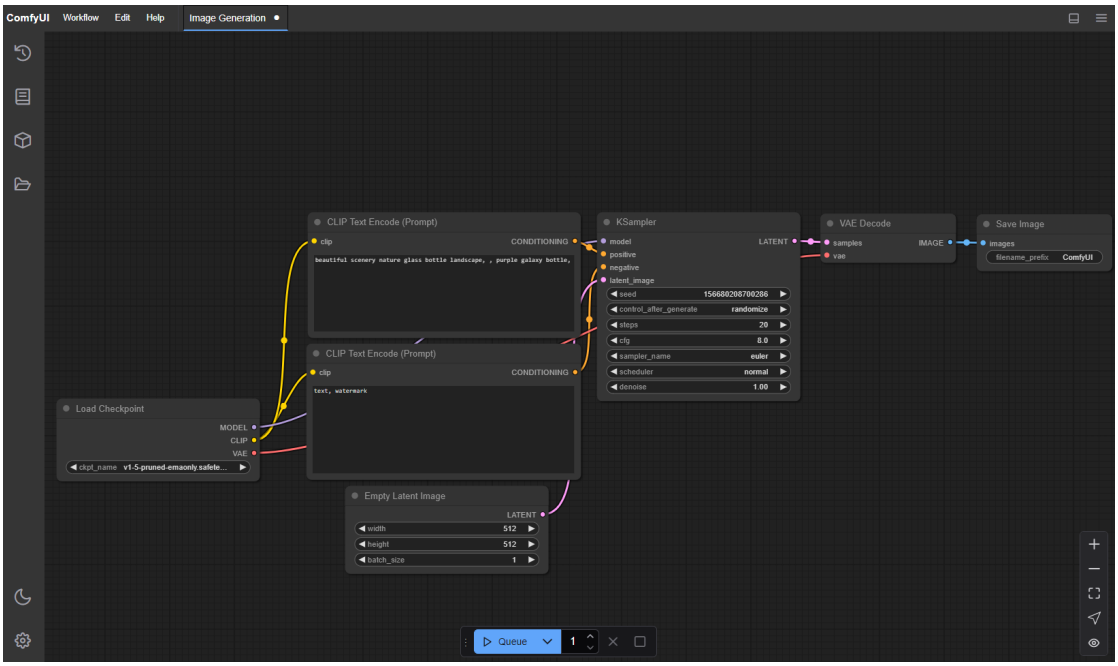


Figura 3.4: *Workflow* base di ComfyUI per generare un'immagine con Stable Diffusion

La forza di questo strumento open-source risiede nella possibilità da parte di qualsiasi sviluppatore di pubblicare i propri nodi personalizzati e renderli disponibili all'intera comunità di utilizzatori. Sebbene il pacchetto di installazione base includa già molti nodi, le funzionalità aggiunte dalla comunità sono infinite e continuano ad aumentare sempre di più, ampliando le possibilità di creazione e sperimentazione. L'installazione di questi nodi è facilitata dal *manager* presente

all'interno dell'interfaccia di ComfyUI che, come mostrato nella figura 3.5, offre diverse funzionalità:

- avere una lista completa di tutti i nodi disponibili, già installati o ancora non presenti nella propria interfaccia (figura 3.6),
- cercare i nodi mancanti in un dato *workflow* e installarli velocemente,
- cercare, scaricare e installare modelli da Civitai,
- installare modelli direttamente da un URL di Git[5],
- cercare le nuove versioni dei nodi installati e aggiornarli,
- aggiornare forzatamente ComfyUI o cambiare versione del programma.

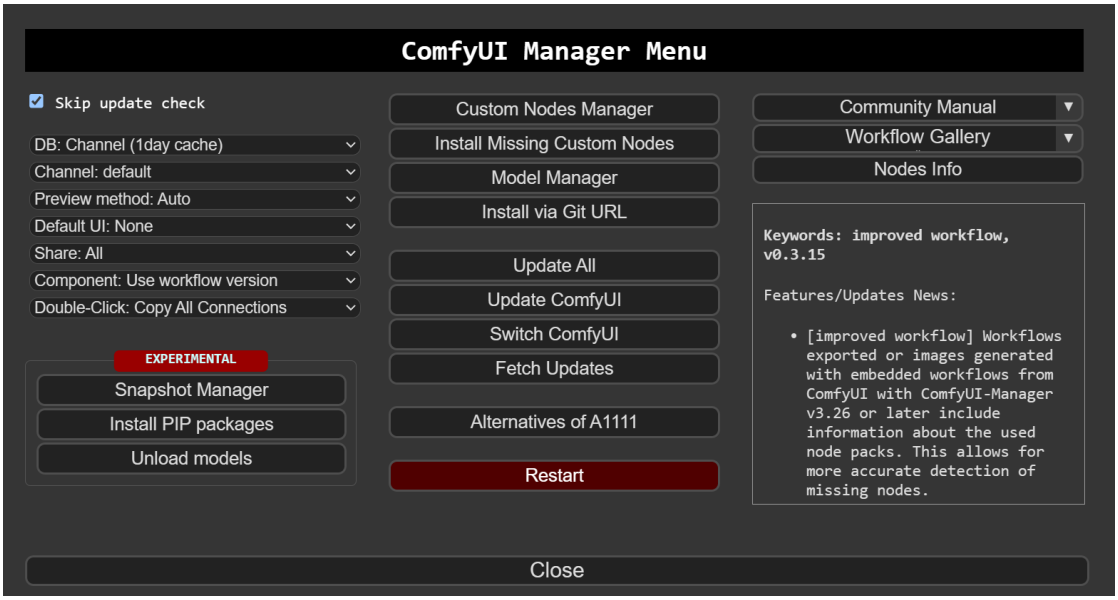


Figura 3.5: Interfaccia del ComfyUI Manager Menu

[5]Sistema di controllo versione distribuito (DVCS, *Distributed Version Control System*) che consente di tracciare le modifiche ai file, coordinare il lavoro tra più sviluppatori e gestire il codice sorgente in modo efficiente.

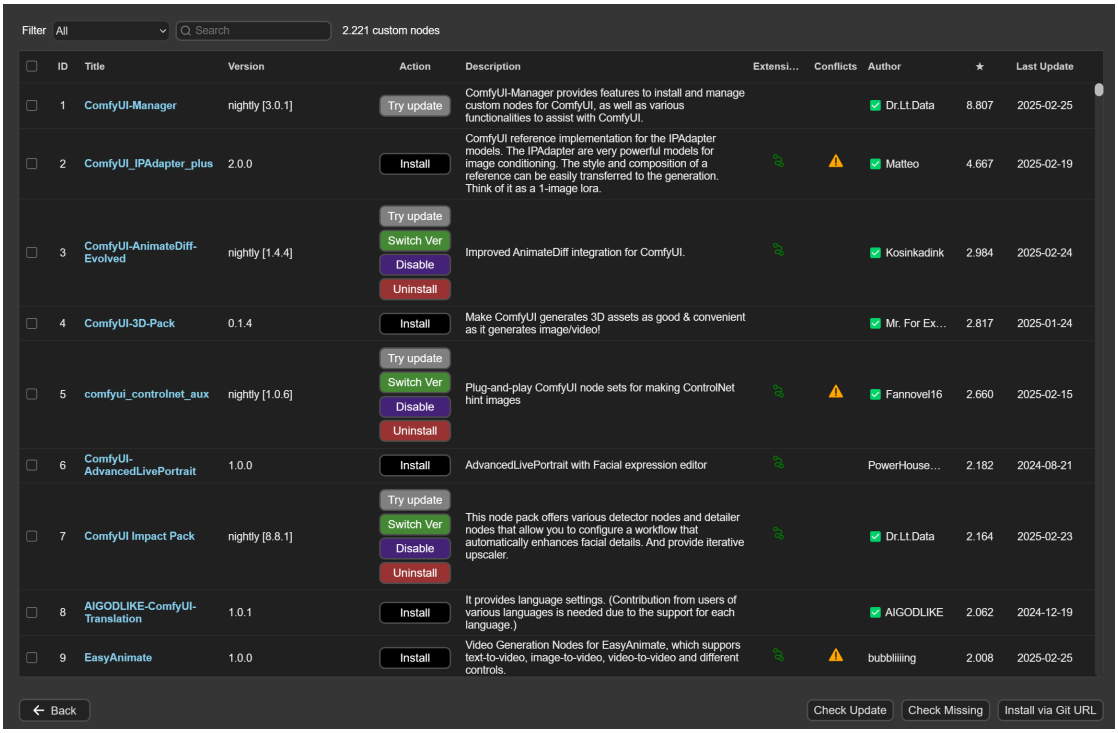


Figura 3.6: Elenco di nodi personalizzati presenti su ComfyUI

I nodi sono uno degli elementi più importanti e caratterizzanti di ComfyUI. Essi contengono delle proprietà o parametri variabili che possono essere modificati manualmente dall'utente o gestiti automaticamente da altri nodi connessi all'input dello stesso. Un parametro interno a un nodo viene chiamato Widget; questo può essere convertito in un Input esterno per controllarne in modo più efficiente il suo valore ed eventualmente condividerlo con altri nodi (figura 3.7). Sebbene ComfyUI sia scritto e si basi interamente sul linguaggio di programmazione Python, che è molto permissivo riguardo ai tipi di dati, ComfyUI è un ambiente fortemente tipizzato. I dati contenuti all'interno di una proprietà devono essere descritti secondo le regole della programmazione informatica (i testi alfanumerici come String, i numeri interi come Integer, e così via) e non è possibile connettere un output di un tipo a un input di un altro [19].

(a) KSampler con `noise_seed` come Widget interno al nodo

(b) KSampler con `noise_seed` come Input esterno al nodo

Figura 3.7: Esempi di utilizzo di KSampler con `noise_seed` come Widget (a) e come Input (b)

Nella figura 3.7 viene mostrato come esempio il nodo KSampler. Questo è uno dei nodi più importanti nella generazione di immagini e video perché è il nucleo della creazione del contenuto. Infatti, il nodo KSampler utilizza il modello fornito e le descrizioni positive e negative per generare il contenuto a partire da un'immagine inizialmente vuota, partendo da un rumore fornito dall'utente o generato casualmente (il concetto di rumore era stato già descritto nel paragrafo 1.2). I parametri da fornire al nodo KSampler sono [20]:

Model Il modello utilizzato per il denoising.

Positive La descrizione positiva del contenuto da generare, ovvero ciò che si vuole vedere.

Negative La descrizione negativa del contenuto da generare, ovvero ciò che non si vuole vedere.

latent_image Un'immagine vuota della dimensione in pixel su cui verrà effettuata la generazione, può essere vista come una tela vuota su cui il modello di IA generativa andrà a disegnare.

seed Il numero casuale utilizzato per creare il rumore.

control_after_generate Permette di modificare il numero di seed dopo ogni *prompt*. Può essere generato casualmente, incrementato, decrementato o mantenuto fisso.

steps Il numero di passaggi utilizzati durante il *denoising*. Più passaggi vengono eseguiti, maggiore sarà l'accuratezza del risultato.

cfg Il valore di *classifier free guidance* che determina quanto il *sampler* deve aderire alla descrizione positiva di partenza. Valori più alti rafforzano la corrispondenza con la descrizione, ma possono ridurre la qualità dell'immagine.

sampler_name Il nome del *sampler* da utilizzare; ne esistono diversi con approcci differenti.

scheduler Il tipo di schedulazione da utilizzare per il processo di *denoising*.

denoise La quantità di rumore nell'immagine che deve essere eliminata. Tendenzialmente il suo valore viene mantenuto a 1.0 per evitare rumore e artefatti.

L'utilizzo di ComfyUI è stato essenziale per l'implementazione di tutti i modelli discussi in seguito.

3.3 Generazione immagini: Stable Diffusion e derivati

Nella scelta del modello per la generazione delle immagini, Stable Diffusion si è dimostrata la soluzione più adattabile alle esigenze del progetto. La sua natura *open source* ha consentito di avere libero accesso al modello completo e a evitare ogni forma di censura o limite nella libertà di espressione [5].

L'utilizzo di Stable Diffusion ha consentito inoltre l'implementazione di un modello LoRA, che consentisse di mantenere la consistenza del personaggio. Questo argomento verrà descritto nel paragrafo 3.4.

La versione 1 del modello era stata pubblicata dal gruppo di ricerca *Computer Vision & Learning* dell'Università Ludwig Maximilian di Monaco, insieme al contributo di Runway, ed è stata il frutto di un progetto chiamato Latent Diffusion. Dalla versione 2 e per le successive XL, 3 e 3.5 il modello è stato acquisito dall'azienda Stability AI [21], che vede la partecipazione di aziende come Amazon Web Services, Nvidia e Intel in qualità di partner per la ricerca [22].

I modelli basati su Stable Diffusion utilizzano modelli di diffusione latente (LDM, *Latent Distribution Models*), noti per la loro capacità di generare immagini di alta qualità e dettaglio. In particolare, questo tipo di modello riesce a

Caratteristica	**DALL·E 3**	**Midjourney**	**Stable Diffusion**
Licenza	Chiuso, proprietario	Chiuso, proprietario	Open-source
Accesso	Solo ChatGPT Plus e Enterprise	A pagamento con diversi piani	Scaricabile e modificabile
Personalizzazione	Limitata, con filtri automatici	Limitata	Elevata (LoRA, fine-tuning)

Tabella 3.1: Confronto tra modelli di generazione immagini: DALL·E 3, Midjourney e Stable Diffusion

ridurre drasticamente le esigenze computazionali necessarie alla generazione dei contenuti, consentendo così il loro impiego anche su macchine di uso comune, senza compromettere la qualità visiva delle immagini generate. Questi modelli sono anche altamente flessibili, grazie all'introduzione di meccanismi di condizionamento basati su *cross-attention*, che permettono al modello di generare immagini mettendo insieme diverse informazioni a partire dalla descrizione di partenza: scrivendo un *prompt* contenente diverse frasi, il modello riesce a suddividere l'immagine in diverse sezioni, per poi associare ogni frase del testo a una specifica area dell'immagine, decidendo dove posizionare ogni descrizione visiva. Così facendo, è maggiormente probabile che il risultato finale soddisfi maggiormente i requisiti descritti nel testo [5].

3.3.1 Modello avanzato: SDXL

Il modello Stable Diffusion ha avuto diverse nuove versioni nel corso degli anni, di cui la più importante è stata denominata Stable Diffusion XL. Questa nuova versione introduce numerose migliorie, grazie principalmente a una rete neurale di dimensioni significativamente superiori (2,6 miliardi di parametri nella versione XL, contro gli 860 mila delle precedenti), che a sua volta amplia anche l'architettura su cui il modello si basa e aggiunge meccanismi di controllo della qualità più sofisticati. Inoltre, l'incremento di complessità consente l'inserimento di descrizioni testuali più articolate, che presentano un livello di dettaglio e di coerenza visiva nettamente superiore. Nella figura 3.8 vengono riportati due esempi di differenza tra i diversi modelli; per ogni *prompt*, vengono mostrate tre immagini casuali del rispettivo modello, generate utilizzando 50 passi, il *sampler* DDIM e un fattore cfg di 8.0.

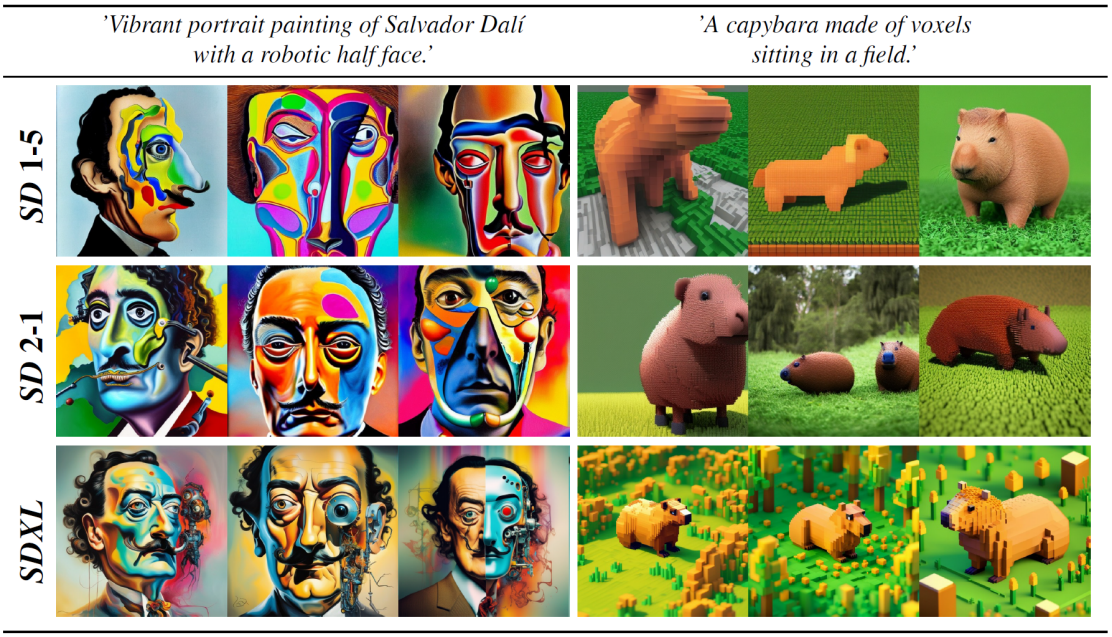


Figura 3.8: Confronto dei risultati di generazione di Stable Diffusion XL rispetto alle versioni precedenti di Stable Diffusion: v1.5 e v2.1

Uno degli aspetti più rilevanti di SDXL è il modo in cui il modello è stato addestrato: i modelli precedenti lavoravano prevalentemente su immagini quadrate di dimensione fissa (tipicamente 512×512 px o 768×768 px); la versione estesa del modello permette di generare immagini in qualsiasi formato si desideri [23]. Anche questo fattore è stato di fondamentale importanza ai fini del progetto: dovendo utilizzare le immagini per la generazione dei video, era necessario che queste fossero nel formato 16:9 tipico del video.

In un contesto di rapida evoluzione nella generazione di contenuti visivi, SDXL offre un'opportunità straordinaria per gli sviluppatori e i ricercatori, consentendo loro di esplorare nuove modalità di espressione e innovazione.

Grazie alle sue ampie possibilità di realismo e coerenza visiva, il modello Stable Diffusion XL è stato utilizzato all'interno del progetto per la generazione di tutte le immagini per poter allenare il modello LoRA del personaggio principale.

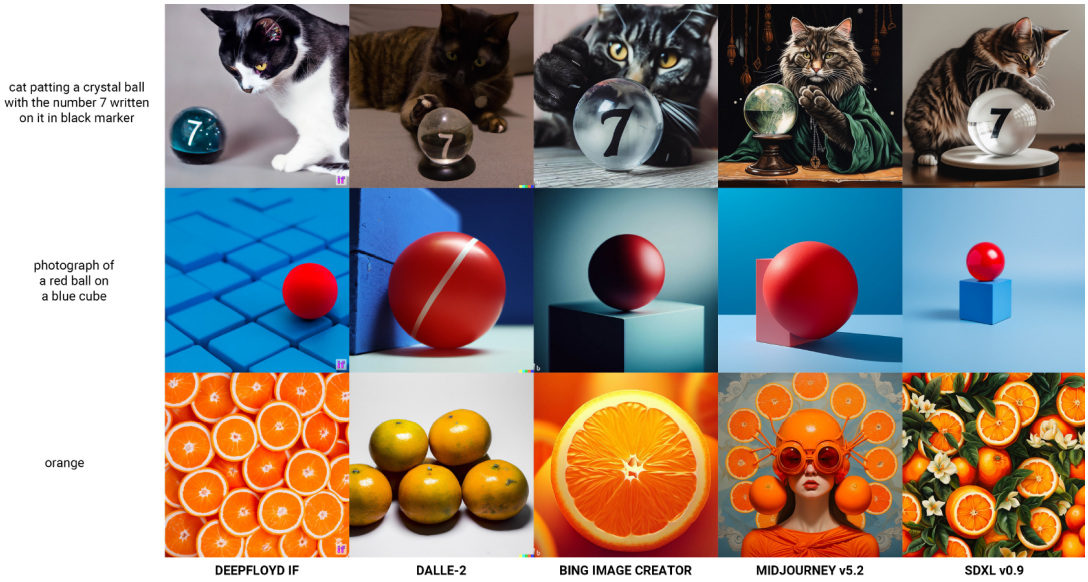


Figura 3.9: Paragone dei risultati di generazione di SDXL rispetto ad altri modelli di generazione: DeepFloyd IF, DALL·E 2, Bing Image Creator, Midjourney v5.2

3.3.2 Modello sperimentale: FLUX.1

Sviluppato dall'azienda Black Forest Labs, FLUX.1 rappresenta il modello open-source attualmente più avanzato e completo. L'azienda propone tre versioni del proprio modello:

FLUX.1 [pro] offre prestazioni all'avanguardia nella generazione di immagini, con attenzione al dettaglio e varietà stilistica, ma è chiuso e a pagamento.

FLUX.1 [schnell] è una versione allegerita e semplificata del modello completo, disponibile gratuitamente e consigliata per contesti di sviluppo locale o di prototipazione rapida.

FLUX.1 [dev] si pone a metà tra le due versioni, è completamente gratuita ma offre qualità e capacità di aderenza ai *prompt* simili a quelle della versione [pro], risultando anche più efficiente dei modelli basati su Stable Diffusion e SDXL [24]; questa è la versione scelta per il progetto.

Nella figura 3.10 viene mostrato un confronto fra queste tre versioni e gli altri principali concorrenti, in un grafico che mostra sulle ascisse il costo in centesimi

per ogni immagine e sulle ordinate il punteggio ELO[6] assegnato a ogni modello dal sito Artificial Analysis. La versione scelta per il progetto è mostrata in un arancione più acceso, mentre il precedentemente citato Stable Diffusion (versione 3 Medium) in verde più chiaro: è possibile vedere, come con un costo in centesimi poco minore, FLUX.1 [dev] dia risultati nettamente migliori e più elevati anche rispetto a modelli più rinomati, come DALL·E 3 e alcune versioni di Midjourney.

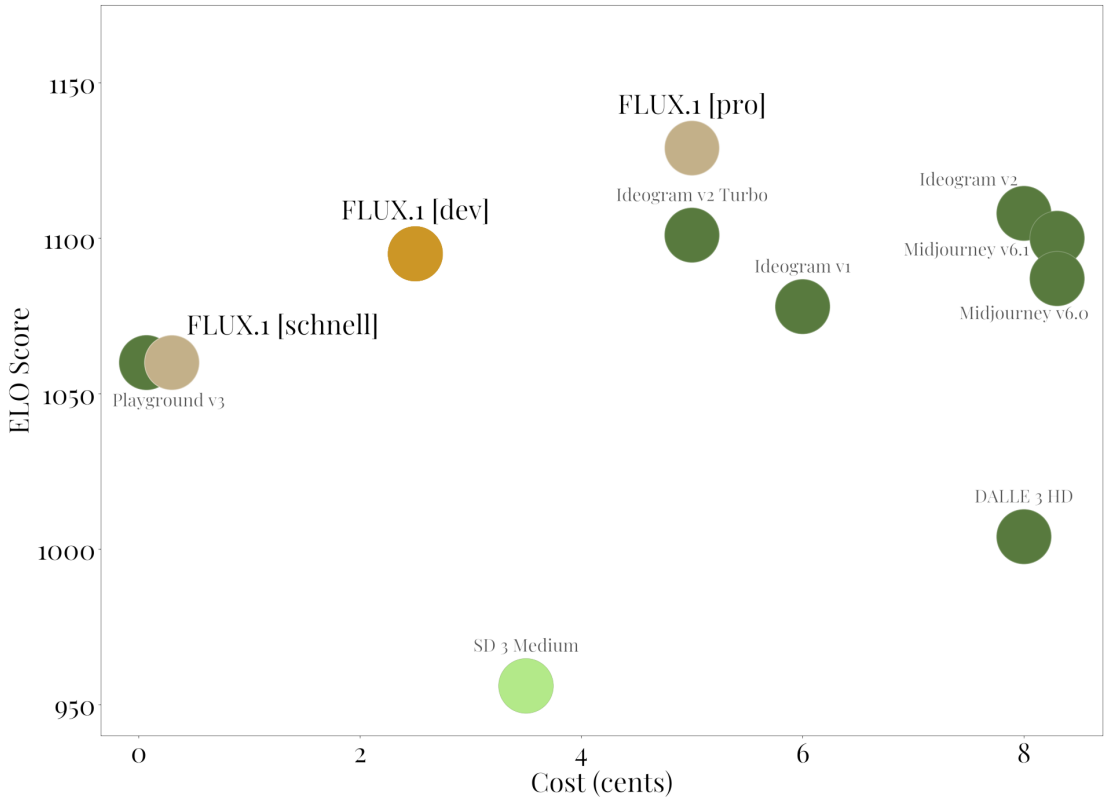


Figura 3.10: Analisi dei punteggi ELO e del costo dei tre modelli FLUX.1 rispetto ai principali concorrenti open-source o a pagamento

In generale, FLUX.1 si è dimostrato la miglior scelta per la realizzazione del cortometraggio, poiché fornisce un bilanciamento unico tra realismo e stile, garantendo un'eccezionale qualità visiva e un'elevata aderenza ai *prompt*. Rispetto a Stable Diffusion XL, FLUX.1 [dev] ha infatti dimostrato una capacità

[6]Misura la qualità percepita delle immagini generate da IA tramite confronti diretti basati sulle preferenze degli utenti. Un punteggio più alto indica una qualità visiva superiore.

superiore di interpretare e tradurre fedelmente le richieste creative, restituendo immagini più precise e coerenti. Nella figura 3.11 queste differenze vengono dimostrate e suddivise in cinque categorie: aderenza alla descrizione, variazione di dimensioni e formato, tipografia, diversità dei risultati, qualità visiva; in ognuno di questi, persino la versione più semplificata di FLUX.1 riesce a superare sia Stable Diffusion 3 che la versione XL. Questo risultato di FLUX.1 è reso possibile da un'architettura che integra blocchi di diffusione multimodali, mantenendo al contempo una complessità computazionale paragonabile a quella di SDXL. Tale caratteristica consente di ottimizzare le prestazioni operative, un fattore particolarmente rilevante nell'ambito della produzione indipendente di questo progetto.

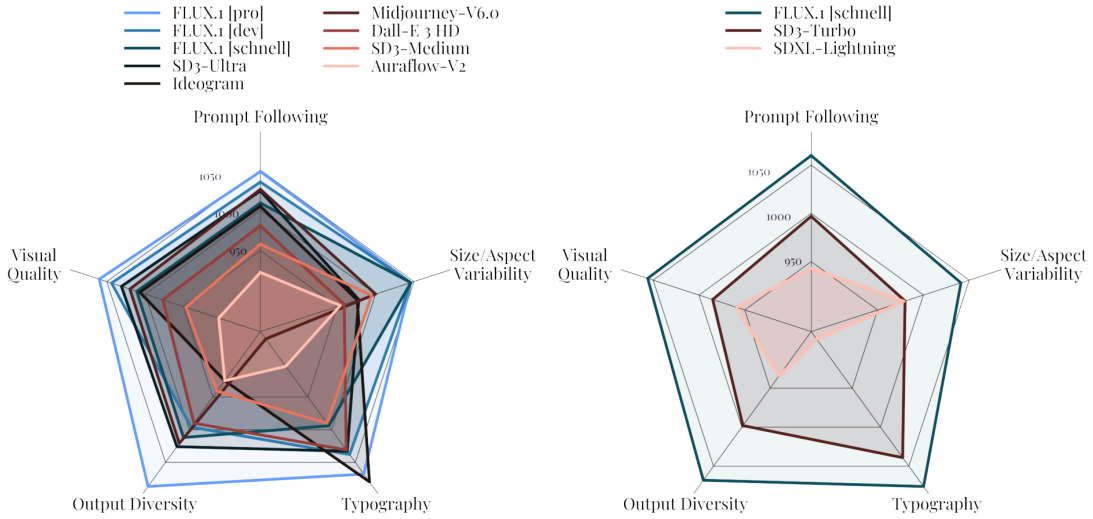


Figura 3.11: Confronto delle performance in cinque categorie dei modelli FLUX.1 rispetto ai principali concorrenti open-source o a pagamento

La figura 3.12 mostra un confronto tra le immagini generate utilizzando SDXL e FLUX.1 [dev], utilizzando due *prompt* diversi. Il primo richiede un ritratto di una giovane donna sorridente in una strada affollata, con in mano un cartello, concentrandosi sulla riproducibilità di frasi testuali. Il secondo, invece, descrive una donna con i capelli rosa con la mano sinistra sopra la testa; in questo caso, l'obiettivo è di analizzare la posa e la generazione delle mani. Per la generazione delle immagini sono stati utilizzati i seguenti parametri: dimensione 1024×1024 px, *sampler* Euler, 20 passi, scheda grafica Nvidia 4090; per ogni *prompt* vengono mostrate le prime tre immagini generate con *seed* casuali [25]. Dalle immagini è evidente che FLUX.1 riesca a garantire una coerenza maggiore

tra *prompt* e risultato visivo: il testo è riprodotto correttamente in tutte e tre le immagini e la posa della donna è sempre corretta, a meno per una differenza tra destra e sinistra. Anche in termini di realismo, le immagini generate da FLUX.1 si adattano maggiormente alle richieste del progetto, mentre quelle di SDXL, seppur buone, danno risultati più artistici e artefatti.

Figura 3.12: Confronto tra due campioni di immagini generate da SDXL e FLUX.1 [dev] utilizzando due *prompt*

Black Forest Labs sta lavorando alla versione Text-To-Video del proprio modello. Quando questa verrà pubblicata, potrebbe rappresentare il nuovo stato dell'arte nella generazione video. L'azienda al momento promette che anche questa nuova versione sarà gratuita, ma per averne la completa certezza è necessario attendere l'effettivo annuncio ufficiale [26].

3.4 Personalizzazione modelli: Kohya_ss

Kohya_ss è un progetto open-source che offre un'interfaccia grafica basata su Gradio[7] per semplificare l'addestramento di modelli per la generazione di immagini, in particolare quelli basati su Stable Diffusion. Questo strumento è stato sviluppato per rendere accessibile a un pubblico più ampio il processo di

[7]Libreria Python per creare interfacce web semplici e interattive per modelli di IA, rendendoli accessibili direttamente nel browser.

configurazione e avvio dell'addestramento dei modelli, riducendo la necessità di competenze tecniche avanzate. Kohya_ss è particolarmente utile per chi lavora con tecniche come *DreamBooth*, *Low-Rank Adaptation* (LoRA) e *Textual Inversion*, che sono ampiamente utilizzate nel campo della generazione di immagini tramite Intelligenza Artificiale [27].

Attualmente, i modelli *DreamBooth* si pongono come i migliori per la personalizzazione, poiché offrono una maggiore fedeltà nella riproduzione di dettagli specifici e concetti unici. Tuttavia, nel contesto di questo progetto di tesi, è stato scelto di lavorare con i LoRA, in quanto richiedono tempi di addestramento più brevi e un minor consumo di risorse computazionali. Inoltre, i LoRA consentono di adattare modelli pre-addestrati a nuovi compiti mantenendo una buona capacità di generalizzazione, evitando che il modello memorizzi eccessivamente i dettagli specifici del dataset di addestramento a discapito della capacità di performare su dati nuovi. I LoRA hanno una struttura semplice che permette di modificare solo una frazione dei parametri del modello, preservando gran parte delle informazioni apprese durante l'addestramento iniziale. Ciò li rende particolarmente adatti per applicazioni in cui è necessario un bilanciamento tra efficienza e prestazioni, come nel caso di modelli di grandi dimensioni o in situazioni in cui non si hanno a disposizione molte risorse (come memoria o potenza di calcolo) [28].

L'interfaccia utente è molto semplificata e strutturata in categorie funzionali, le cui più importanti sono quelle relative al modello e alla configurazione dell'addestramento (figura 3.13). I dettagli delle varie schede saranno descritte con più dettaglio nel paragrafo 3.4, in cui si descriverà anche il procedimento di configurazione e creazione di un modello LoRA.

Sono disponibili anche degli strumenti per agire direttamente sui LoRA precedentemente creati, consentendo l'estrazione, la fusione o l'ottimizzazione di modelli specifici (figura 3.14). Ad esempio, Extract DyLoRA permette di isolare una rete dinamica LoRA da un modello pre-addestrato, mentre Merge LoRA (SVD) utilizza la decomposizione ai valori singolari per combinare reti in modo efficiente.

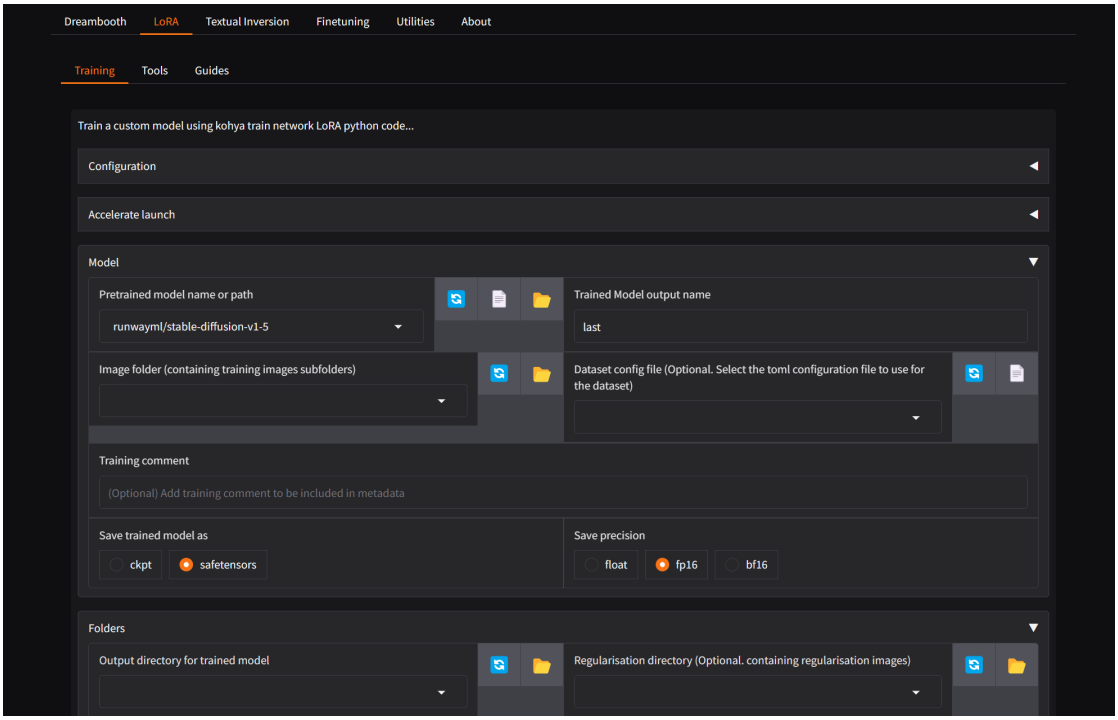


Figura 3.13: Interfaccia di Kohya_ss per la configurazione di un modello LoRA

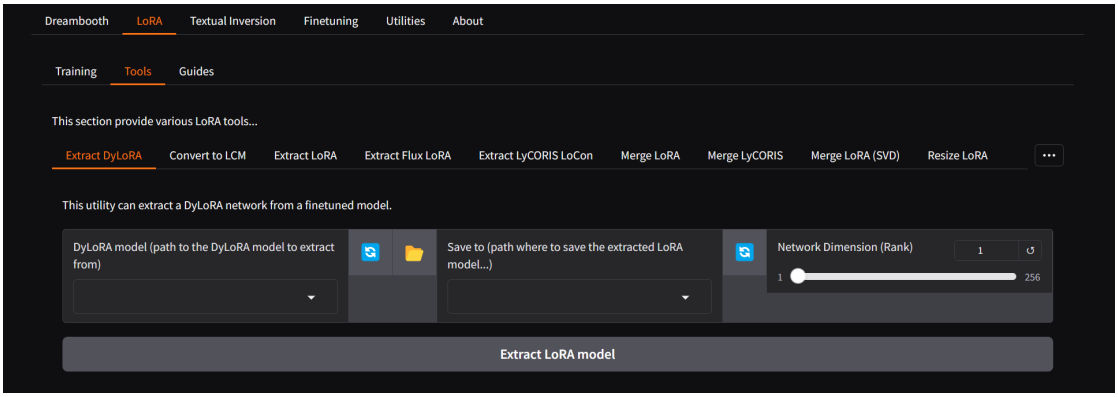


Figura 3.14: Strumenti di modifica o analisi dei modelli LoRA forniti da Kohya_ss

3.5 Generazione video: CogVideoX

La fase di selezione dello strumento più adatto per la generazione video è stata la più lunga e complessa, attraversando diverse fasi di sperimentazione e

valutazione. In una prima fase, sono stati testati strumenti come Stable Diffusion Video e AnimateDiff, che avevano mostrato un buon potenziale iniziale, ma presentavano ancora forti limiti in termini di stabilità, coerenza temporale e qualità visiva complessiva. Parallelamente, sono stati esplorati anche altri modelli sperimentali che, tuttavia, si sono rivelati talmente instabili e poco supportati da risultare presto obsoleti e abbandonati dalla stessa comunità di sviluppo. Dopo questa fase di esplorazione, la svolta è arrivata con l'adozione di CogVideo, selezionato come strumento principale nel mese di settembre 2024, in concomitanza con il rilascio della versione CogVideoX-5B il 27 agosto 2024, che introduceva significativi miglioramenti rispetto alle versioni precedenti, offrendo una maggiore stabilità nella generazione e una qualità visiva nettamente superiore. A partire da quel momento è iniziato il vero e proprio lavoro di generazione dei video per il progetto.

Come già discusso nel paragrafo 1.2, il settore della generazione video ha vissuto un'evoluzione estremamente rapida negli ultimi due anni, con modelli come Runway Gen-3 Alpha e Sora che hanno introdotto tecniche sofisticate per garantire maggiore coerenza e fedeltà visiva. Tuttavia, questi strumenti, pur rappresentando lo stato dell'arte, presentano alcune limitazioni dal punto di vista dell'accessibilità tecnologica e del costo di utilizzo, essendo entrambi servizi chiusi e proprietari. In questo contesto, CogVideo si distingue come uno dei modelli open-source più avanzati, rappresentando una soluzione interessante proprio per la sua disponibilità pubblica e per l'elevato livello qualitativo che riesce a garantire.

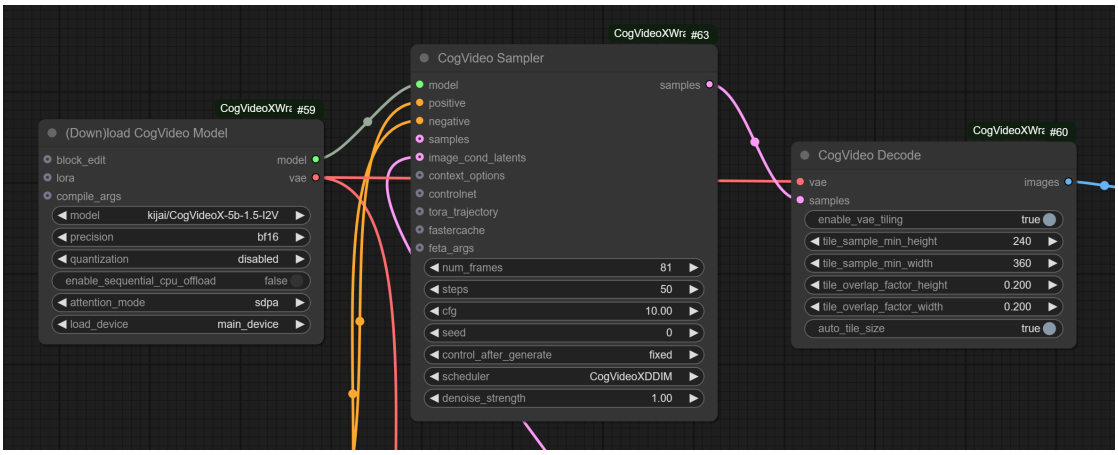


Figura 3.15: Nodi essenziali da inserire in un *workflow* per l'utilizzo di CogVideoX. Interfaccia di ComfyUI

Il suo utilizzo è integrato all'interno dell'interfaccia ComfyUI. Sono disponibili dei *workflow* base già completi di tutti i nodi necessari alla generazione di tipo Text-To-Video o Image-To-Video, ampliabili o modificabili a seconda delle proprie necessità. Nella figura 3.15 sono mostrati i tre nodi principali per la generazione di un video, da cui è possibile scegliere e scaricare direttamente da ComfyUI il modello di generazione video preferito (figura 3.16), selezionare il numero di *frame* video da generare e lo *scheduler* da usare.

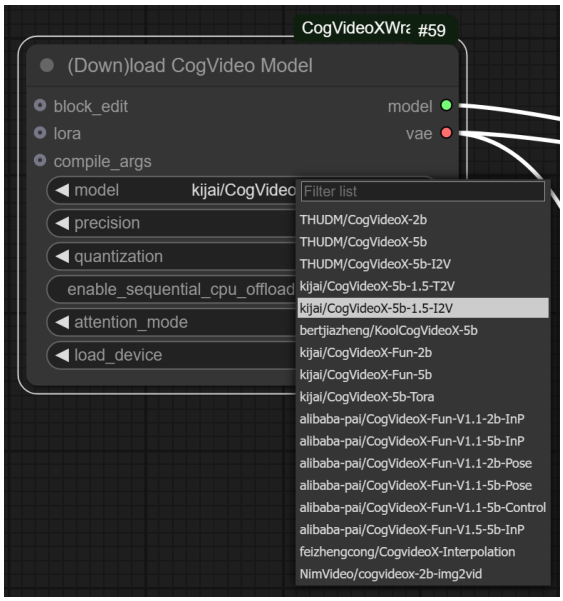


Figura 3.16: Elenco dei modelli CogVideoX disponibili per l'utilizzo, scaricabili direttamente dal parametro model del nodo Down(load) CogVideo Model. Interfaccia di ComfyUI

Uno degli aspetti più rilevanti di CogVideo, anche in relazione agli obiettivi di questa ricerca, è l'introduzione di una strategia di addestramento gerarchico a multi-frame rate, che consente al modello di migliorare significativamente la coerenza temporale tra i frame generati, mantenendo una maggiore fedeltà rispetto al testo fornito in input. Questo approccio, che combina un controllo esplicito sulla dinamica delle scene con una rappresentazione semantica più stabile, si è rivelato particolarmente efficace nel generare sequenze fluide e naturali, un aspetto fondamentale per applicazioni creative e narrative nel contesto della produzione audiovisiva [4].

Con i nuovi aggiornamenti e con la versione più recente CogVideoX il modello ha superato, sia in termini di metriche automatiche che di valutazioni umane, molti dei modelli open-source precedentemente disponibili. Proprio questa

combinazione tra qualità del risultato, accessibilità e trasparenza del modello, ha reso CogVideoX una scelta ideale come punto di riferimento per questo progetto di ricerca, evidenziando come l'adozione di modelli *transformer* su larga scala stia progressivamente ampliando le possibilità creative e produttive nel settore dei contenuti multimediali generati tramite intelligenza artificiale [29].

Nei mesi successivi, il modello è stato ulteriormente aggiornato con l'arrivo di CogVideoX1.5-5B e CogVideoX1.5-5B-I2V, rilasciati nel novembre 2024. Queste versioni hanno ampliato le possibilità tecniche, introducendo il supporto al formato 16:9 (inizialmente la generazione era limitata al formato 4:3) e aumentando la durata massima dei video generabili fino a 10 secondi a 8 fps, rispetto al limite precedente di 6 secondi a 8 fps. Il progetto si è progressivamente aggiornato e adattato a queste nuove versioni, sfruttando al massimo le nuove capacità offerte dal modello e migliorando così la qualità complessiva dei risultati finali.

Caratteristica	**Sora**	**Runway (Gen-2/Gen-3)**	**CogVideoX**
Licenza	Chiuso, proprietario	Chiuso, SaaS (*Software as a Service*)	Open-source (basato su CogVideo)
Accesso	Non ancora pubblico	Accessibile via browser con account Runway	Disponibile su arXiv per ricerca
Modello di generazione	Diffusione avanzata con trasformatori	Diffusione latente + Video-to-Video	Diffusione + Expert Transformer
Input supportati	Text-To-Video	Text-To-Video Image-To-Video Video-To-Video	Text-to-Video Image-To-Video
Qualità video	Alta (risoluzione elevata e dettagli realistici)	Buona (coerenza migliorata rispetto a Gen-1)	Media (coerenza temporale migliorata)
Durata massima	Video lunghi e fluidi	Brevi clip (fino a pochi secondi)	Clip brevi con coerenza temporale migliorata
Filtri	Controllo avanzato su contenuti sensibili	Moderazione automatica	Nessun filtro attivo (open-source)

Tabella 3.2: Confronto tra modelli di generazione video: Sora, Runway e CogVideoX

3.6 Altre tecnologie freemium e premium

Per alcuni scopi specifici o per alcuni compiti troppo complessi da gestire con gli strumenti precedentemente citati, sono stati utilizzati anche altri strumenti in parte o totalmente chiusi, talvolta a pagamento, che vengono riportati di seguito.

3.6.1 Eleven Labs

Eleven Labs rappresenta uno degli strumenti allo stato dell'arte nel campo della vocalizzazione e offre funzionalità per la conversione di testi in voce, il doppiaggio di voci in lingue diverse da quella originale e per il cambiamento completo della voce [30]. Nel contesto del progetto, questo strumento è stato utilizzato per la generazione della voce del partigiano, che racconta con una voce fuori campo le proprie vicende di vita.

Un primo approccio per la generazione di tale voce è stato di tipo Text-To-Voice in cui, fornito il testo del discorso da utilizzare per il video, lo strumento di IA Generativa restituiva un file audio contenente il discorso. Questa soluzione presentava alcuni evidenti problemi di intonazione, intenzione e velocità del parlato, dovuti alla cattiva interpretazione del testo da leggere da parte di Eleven Labs, oltre a una mancanza di informazioni sul contesto storico e culturale da rappresentare, che non possono essere inserite durante la fase di configurazione.

Successivamente, è stata effettuata un'altra prova utilizzando il metodo Speech-To-Speech. Dopo aver registrato la voce fuori campo interpretata da Stefano Sburlati, lo strumento di *respeech* ha cambiato la voce dell'interprete camuffandola e, con l'utilizzo di alcuni strumenti di modifica tono e velocità proposti dalle nuove versioni di Eleven Labs [31, 32], si è riusciti ad arrivare a un risultato soddisfacente. Il modello di generazione scelto è *Eleven Multilingual 2* che, tra le altre cose, riesce ad elaborare l'intenzione emotiva del testo [31], con la libreria vocale Roberto Riva, descritta come "Voce maschile anziana con sfumature italiane, perfetta per la narrazione". Nella figura 3.17 vengono riportati i parametri utilizzati per la configurazione della voce scelta:

Stabilità al 30% per ridurre la monotonia e aggiungere variazioni di tono su segmenti di testo lunghi;

Similarità al 40% per rendere la voce camuffabile da quella di Stefano Sburlati ma ridurre al tempo stesso gli artefatti che valori alti di questo parametro causano;

Esagerazione dello stile al 50% per aumentare l'intenzione e la profondità del dialogo rispetto alla voce originale (selezionando valori maggiori il sistema restituirà un messaggio in cui avvisa che valori superiori al 50% potrebbero causare instabilità).

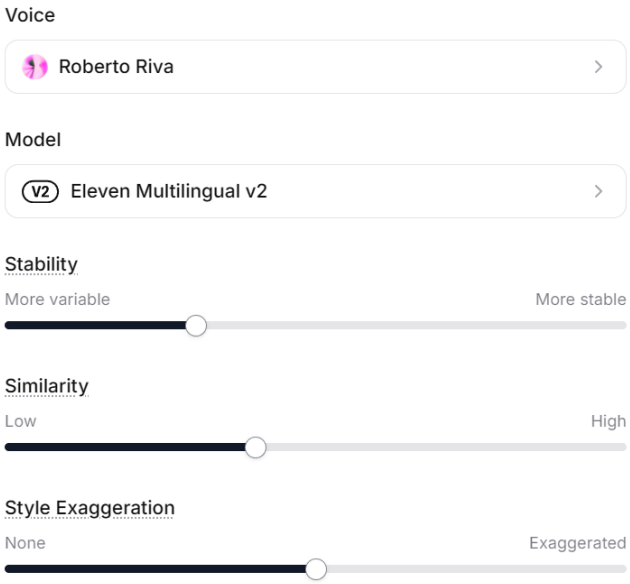


Figura 3.17: Parametri per il cambiamento della voce con Eleven Labs

3.6.2 Hedra

Hedra è una piattaforma *freemium* di IA Generativa per la creazione di video con l'integrazione di avatar personalizzati, animazioni sincronizzate con l'audio e sintesi vocale avanzata. Grazie a queste funzionalità, con l'utilizzo di questo strumento è possibile sincronizzare i movimenti labiali dei personaggi sincronizzati con le parole da loro pronunciate [33].

Nella scena 7 della sceneggiatura del trailer realizzato, il partigiano, ormai anziano, parla guardando in camera riflettendo sui propri ricordi e la propria storia. Il solo utilizzo di CogVideo non era sufficiente a dare una resa realistica del movimento labiale del personaggio col discorso pronunciato, per cui l'utilizzo di Hedra è stato fondamentale per questo compito. I passi preliminari per l'utilizzo di Hedra sono stati due: la generazione dell'immagine di riferimento, utilizzando il *workflow* descritto nel paragrafo 4.3 con l'utilizzo di FLUX.1 e del modello LoRA del partigiano, e la creazione della voce del partigiano con

l'utilizzo di Eleven Labs. Successivamente, utilizzando l'interfaccia mostrata nella figura 3.18, bisogna selezionare il modello di generazione (oltre a quello proprietario sono presenti anche i più conosciuti Hunyuan, Kling e Veo 2), il formato, la definizione e una breve descrizione di quello che dovrebbe accadere nel video generato.

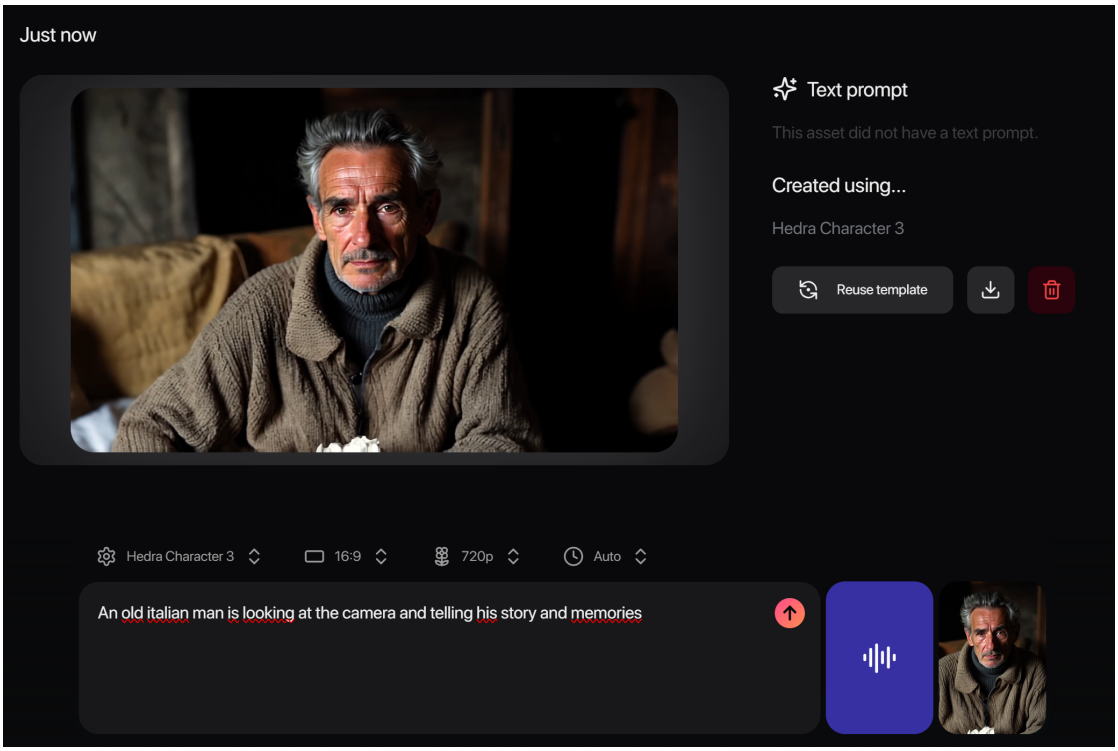


Figura 3.18: Interfaccia di Hedra per la sincronizzazione video-audio

3.6.3 Suno

Suno è una piattaforma di generazione musicale con l'utilizzo dell'Intelligenza Artificiale, popolare per la sua capacità di creare musica personalizzata a partire da descrizioni testuali [34].

L'integrazione di uno strumento come Suno nel contesto del progetto è dovuta alla necessità di creare una colonna sonora per il trailer, che nei mesi di produzione del progetto gli strumenti open source non erano ancora in grado di realizzare. Il suo utilizzo per la generazione di un brano è molto semplice: è necessario inserire una breve descrizione del brano desiderato, indicandone il genere, lo stile e l'ambientazione desiderata; nel caso della composizione di questa colonna sonora, si deve anche selezionare il pulsante per la composizione

di una traccia strumentale, ovvero senza testo. Nella figura 3.19 viene mostrata la schermata descritta.

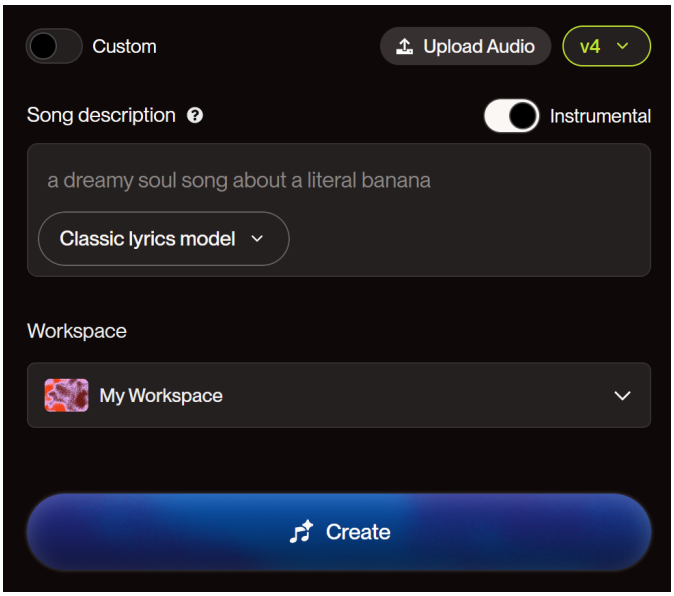


Figura 3.19: Scheda di configurazione di un brano musicale utilizzando Suno

Capitolo 4

Produzione

Il quarto capitolo è dedicato interamente alla descrizione dei metodi e delle procedure utilizzate per la produzione del trailer della serie. Dopo un'iniziale studio e valutazione di diversi metodi di lavoro, è stata definita una *pipeline* stabile e robusta, che consentisse di portare avanti il progetto resistendo ai continui e repentini cambiamenti dell'industria dell'Intelligenza Artificiale Generativa.

4.1 Definizione della pipeline

Come già descritto nei capitoli precedenti, in particolar modo nel paragrafo 1.2.1, una delle fasi più importanti per la realizzazione di questo lavoro è stata quella relativa alla scelta della pipeline di produzione da seguire. Inizialmente, si credeva che imitare e seguire il flusso di lavorazione delle grandi piattaforme a pagamento fosse la strada corretta da percorrere: studiare accuratamente le descrizioni testuali da usare come *prompt*, modellare e costruire un personaggio a partire dalle foto reali del partigiano Cesare Alvazzi del Frate, costruire un *workflow* ComfyUI che restituisse direttamente il video finale, o per lo meno le singole clip che sarebbero state poi montate manualmente.

Si è dovuto fare presto i conti con la realtà: lavorando con piattaforme open-source, non ampiamente e accuratamente curate, o comunque non finalizzate a un utilizzo di alto livello, non è così semplice come aprire Runway e realizzare un video di un soldato in battaglia su una montagna.

Una delle prime prove effettuate prevedeva la generazione di immagini con il metodo Text-To-Image utilizzando alcuni modelli basati su Stable Diffusion 3 o

XL. Queste immagini però erano poco accurate e le ambientazioni rappresentate non erano facilmente paragonabili a quelle del contesto italiano in cui si sono svolte le vicende. Era infatti necessario fornire descrizioni molto accurate e complesse al modello per poter creare delle scene che si avvicinassero a ciò che si voleva raggiungere. Anche il partigiano, per quanto accurate e precise fossero le descrizioni, aveva sembianze e connotati sempre diversi. Ad esempio, anche inserendo la descrizione "*brown short hair*", i capelli non erano mai uguali, a volte più lunghi, a volte più corti, con la riga a sinistra o a destra, ecc.

Una buona soluzione a questo problema è stata l'introduzione di modelli LoRA pre-allenati e basati su immagini selezionate in modo da rispecchiare il risultato finale.

In conclusione, la scelta della pipeline di lavoro definitiva è stata un lavoro di tipo *try-and-error*, in cui è stato fondamentale fare una prova e aggiungere man mano un nuovo pezzo alla struttura complessiva, fino a formare una grande catena di strumenti per arrivare alla generazione di un video di buona qualità e resa.

4.2 Generazione del LoRA

Per la creazione del modello che fornisse il personaggio protagonista della serie, è stato utilizzato lo strumento Kohya_ss, già descritto al paragrafo 3.4. Per il corretto utilizzo dello strumento e per ottenere un modello che fosse abbastanza stabile ed efficiente, è stato necessario individuare una serie di almeno 30 immagini da sottoporre al modello come riferimento per il personaggio e lo stile desiderati, oltre a un campione più ampio di figure più generiche a cui fare riferimento, secondo un fattore moltiplicatore 100× per definire il personaggio finale.

La selezione delle immagini da fornire al modello per influenzare il personaggio è stata effettuata utilizzando delle catture a due lungometraggi sulla resistenza partigiana durante la Seconda guerra mondiale:

Il partigiano Johnny Film del 2000 diretto da Guido Chiesa, il cui protagonista è interpretato da Stefano Dionisi. Questo lungometraggio si contraddistingue per dei buoni primi piani del personaggio e una fotografia cupa e ben contrastata, soprattutto per le ambientazioni interne o notturne.

I piccoli maestri Film del 1997 diretto da Daniele Luchetti, il cui protagonista è interpretato da Stefano Accorsi. La scelta di questo lungometraggio fornisce al modello delle buone ambientazioni di sfondo, oltre a dare dei toni più neutri e colorati alle scene esterne. Inoltre, l'unione dei volti dei due personaggi consente di effettuare una sperimentazione in merito alla creazione di nuovi personaggi originali. Di questo argomento si è discusso meglio nel paragrafo 1.3 dedicato alle etiche di utilizzo delle tecnologie di IA Generativa e se ne discuterà di nuovo nel seguito.

Da questi due lavori sono state tratte 55 immagini che raffiguravano uno o l'altro attore in primo piano. È stato fondamentale selezionare immagini che avessero anche uno sfondo che descrivesse le ambientazioni e lo stile che si volevano ottenere nella serie finale e dei costumi che rispecchiassero quelli dell'epoca e che il partigiano Libero dovesse indossare. A ognuna di queste 55 immagini, è stato abbinato un file di testo `.txt` generato automaticamente da uno degli strumenti di Kohya_ss che contenesse una breve descrizione della stessa; ad esempio, per la prima immagine della figura 4.1 la descrizione generata è "*a man in a trench coat and tie looking at the camera with a serious look on his face*".

Successivamente, è stato necessario generare un numero di immagini corrispondenti a quello scelto precedentemente, moltiplicato per il fattore 100×

Figura 4.1: Campione di 10 immagini su 55 totali selezionate come base di partenza per la generazione del modello LoRA del partigiano

menzionato precedentemente, da fornire al modello. La creazione di queste immagini è stata effettuata con Stable Diffusion XL, che ha consentito di creare un *batch* di generazione rapido e automatico, e il workflow base fornito da ComfyUI per la creazione di immagini col metodo Text-To-Video, già mostrato nella figura 3.4. Sono state generate 5.550 immagini contenenti un primo piano di una figura maschile generica, utilizzando il prompt "*an italian 25-year-old man, brown short hair, year 1940*". Nella figura 4.2 viene mostrato un campione ristretto delle immagini prodotte.

Figura 4.2: Campione di 15 immagini su 5.500 totali di una figura maschile generica generate come confronto per la generazione del modello LoRA

Dopo aver ottenuto queste immagini, è possibile configurare la generazione del modello attraverso Kohya_ss. Aprendo la sezione dedicata alla generazione di modelli LoRA, per prima cosa bisogna andare nel pannello *Dataset Preparation*, da cui si selezionano le immagini precedentemente citate, come si può vedere nella figura 4.3. La combinazione di Instance prompt e Class prompt, definita come *trigger word*, servirà a definire univocamente il modello LoRA durante la generazione delle immagini; ovvero, inserendo nel *prompt* il testo "*p4rtigiano man*", il modello saprà che per la generazione di quel personaggio dovrà utilizzare le informazioni contenute nel LoRA. La parola "partigiano" viene appositamente camuffata con la cifra '4' al posto della prima 'a' per rendere univoca la connotazione e non essere scambiata come traduzione generica del termine. Il pulsante *Prepare training data* si occupa di creare le cartelle di destinazione del modello nella posizione scelta e di copiare le immagini di addestramento e regolazione in delle apposite cartelle. Invece, il pulsante sottostante *Copy into respective fields* inserisce i percorsi delle immagini negli altri pannelli in cui è necessario dichiararli.

Il passo successivo consiste nel selezionare il modello base di riferimento per la generazione e il formato che il modello LoRA dovrà avere. Come mostrato nella

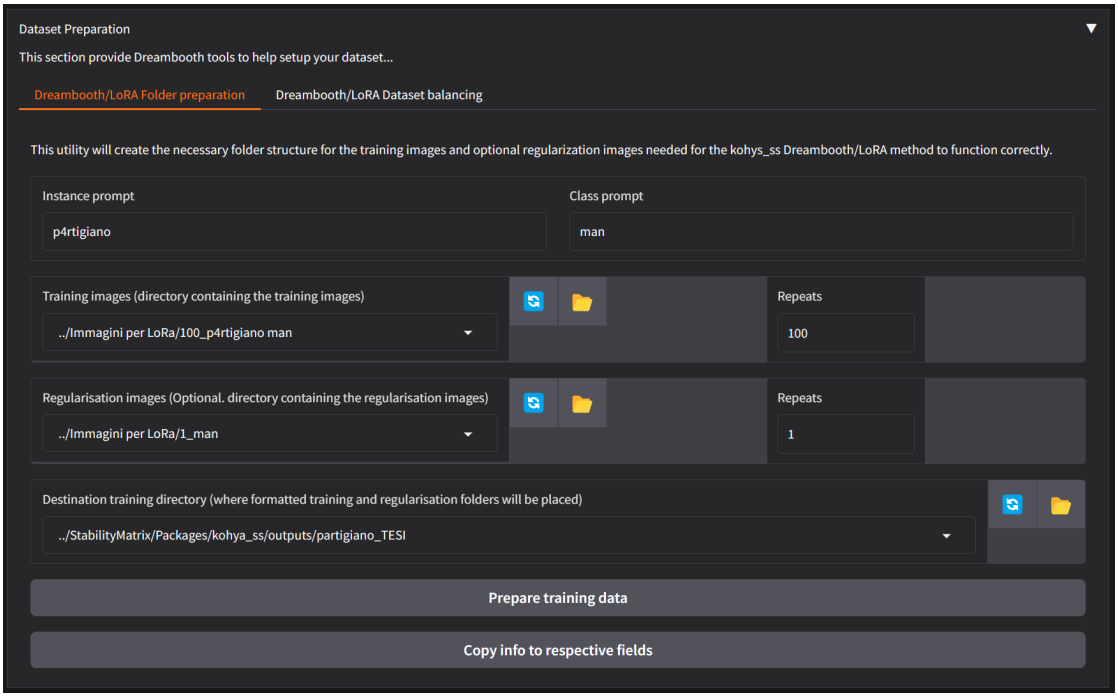


Figura 4.3: Sezione *Dataset Preparation* di Kohya_ss

figura 4.4, il modello scelto è FLUX.1 [dev], lo stesso con cui successivamente saranno generate le immagini[8] e già descritto nel paragrafo 3.3.2. Il formato scelto per il modello è .safetensor, un nuovo formato binario ottimizzato per il caricamento sicuro ed efficiente di modelli di IA, che consente di evitare vulnerabilità come l'esecuzione di codice malevolo (presente nei file .ckpt, che possono contenere codice Python). Invece, per quanto riguarda la precisione di calcolo, bf16 (bfloat16) è un formato numerico a 16 bit che mantiene la gamma di valori del float32, ma con meno precisione nei decimali; rispetto a fp16 (float16), è più stabile nei calcoli e riduce errori nei modelli di IA. Il nome scelto per il modello prescinde dai dati inseriti per la generazione e serve solo ai fini di corretta archiviazione e utilizzo; nel caso dell'esempio mostrato, verrà creato il file partigiano_TESI.safetensor.

L'ultima e più complessa sezione da configurare è *Parameters*, relativa per l'appunto ai parametri più specifici per la generazione del modello LoRA. Vengono descritti di seguito i parametri più rilevanti e per cui sono stati

[8]È necessario che il modello su cui si basa il LoRA sia uguale a quello con cui si generano le immagini; generalmente non è consentita la combinazione di modelli base diversi.

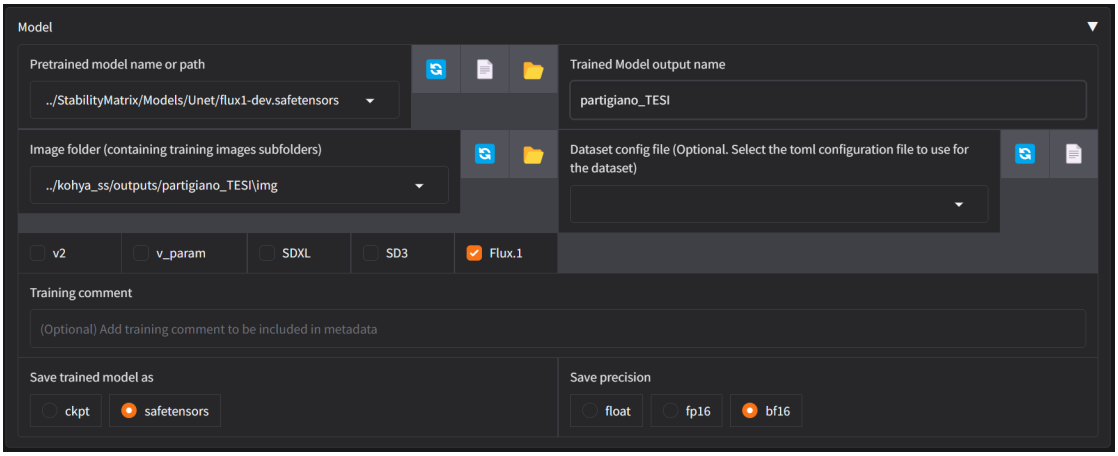


Figura 4.4: Sezione *Model* di Kohya_ss

eseguiti il maggior numero di prove, mostrati anche nella figura 4.5. Durante l'addestramento di un modello, le immagini non vengono elaborate una per volta ma sono raggruppate in *batch*. Il numero di *Epoch* indica il numero di iterazioni attraverso tutte le *batch* che la rete neurale esegue per la generazione del modello. Ad esempio, se le immagini vengono suddivise in cinque *batch*, durante una *Epoch* il numero di iterazioni sarà uguale al numero di *batch*, ovvero cinque. Queste ripetizioni e iterazioni servono a minimizzare l'errore finale, ovvero minimizzare la funzione di perdita del modello; per ogni iterazione, la rete neurale regola i propri parametri per avvicinare il risultato al minimo locale della curva, ogni volta di un determinato passo la cui grandezza è determinata dal valore del *Learning rate*, un parametro compreso tra zero e uno; un valore troppo alto potrebbe rendere difficile trovare il punto di minimo perché la rete cambia notevolmente tra un passo e l'altro, mentre un valore troppo basso potrebbe richiedere più tempo e un numero più elevato di *Epoch* per arrivare al minimo locale [35].

In generale, una buona regola da tenere in mente è che più è alto il valore di Epoch, più basso dovrà essere quello di Learning rate. Ad esempio, è possibile impostare Epoch = 1, Learning rate = 0,0001, ma il risultato di questi valori, a fronte di una generazione più rapida, potrebbe essere poco accurato e dare dei risultati di scarsa qualità, sia per quanto riguarda la diversità delle immagini tra *prompt* e generazioni diverse, sia per quanto riguarda l'originalità del contenuto. Il miglior compromesso trovato dopo diverse prove, e che ha funzionato al meglio sia con il modello FLUX.1 [dev] che con il *dataset* precedentemente preparato, è stato di Epoch = 30, Learning rate = 0,00003: il valore alto di Epoch ha

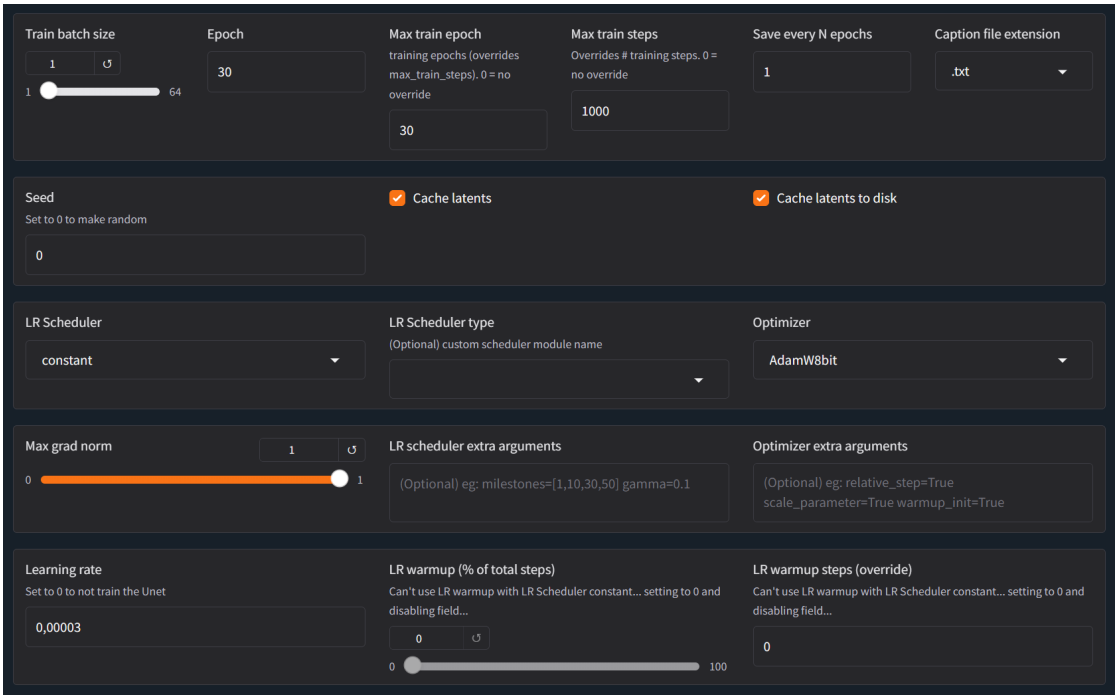


Figura 4.5: Sezione *Parameters* di Kohya_ss relativa alla configurazione dei valori di Epoch e Learning rate

consentito la maggior iterazione delle immagini, aumentando la stabilità e la precisione del modello finale, mentre il valore più basso di Learning rate ha evitato che si verificassero errori di eccessiva accuratezza e *overflow*[9].

Lo *scheduler* scelto è di tipo costante: ciò significa che il learning rate non cambia tra un'iterazione e l'altra seguendo una funzione matematica (altri tipi sono coseno, lineare, polinomiale, ecc.). Invece, l'*optimizer* scelto è AdamW8bit, la versione ridotta a 8 bit del più preciso AdamW: dato il numero elevato di Epoch, si è scelto di ridurre la precisione di questo parametro per diminuire i tempi di generazione ed evitare gli errori di *overflow* poc'anzi menzionati.

Per la generazione di modelli LoRA basati sul modello base FLUX.1 e i suoi derivati, è necessario compilare ulteriori campi, mostrati nella figura 4.6. FLUX.1,

[9]Si verifica quando un numero supera il massimo valore rappresentabile, portando a risultati infiniti (Inf, -Inf) o errori di calcolo, un problema comune nei modelli di IA , dove i valori dei pesi e dei gradienti possono crescere rapidamente durante l'addestramento del modello.

infatti, richiede la combinazione del modello base ad altri pacchetti: un *Variational Autoencoder* (VAE), un *Contrastive Language-Image Pre-training* (CLIP) e un *Text-to-Text Transfer Transformer* (T5); in questo caso il pacchetto T5-XXL è una variazione estesa a 11 miliardi di parametri dell'originale T5 (che ne aveva 76 milioni), sviluppati entrambi da Google.

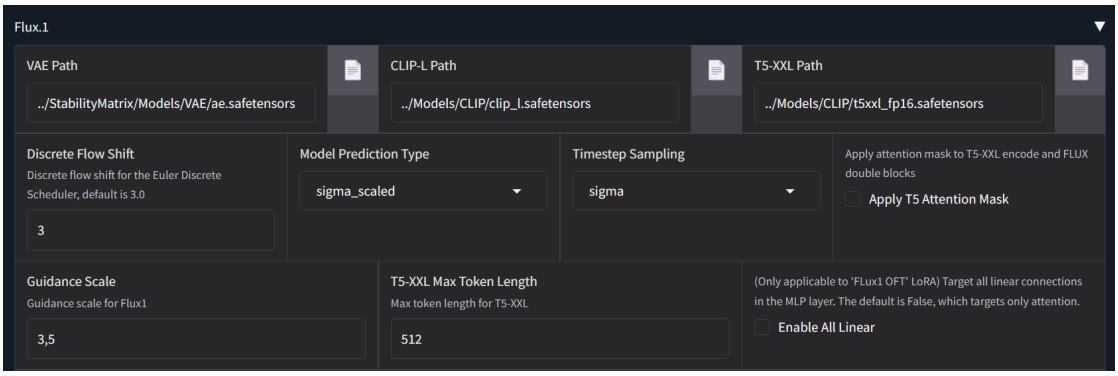


Figura 4.6: Sezione *Parameters - Flux.1* di Kohya_ss

Nella figura 4.7 viene mostrata l'influenza del modello LoRA sulla generazione delle immagini con FLUX.1 [dev]. La differenza tra le due versioni è evidente sia dal punto di vista del realismo visivo che della coerenza narrativa. Nella prima immagine, ottenuta con il solo modello base, l'illuminazione è ben bilanciata, ma il volto appare più levigato e generico, con tratti meno caratterizzati e un'estetica che risulta quasi moderna. Anche i vestiti presentano texture più uniformi e meno dettagliate, riducendo la credibilità storica della scena. Al contrario, nella seconda immagine, generata con l'aggiunta del LoRA personalizzato, l'atmosfera diventa più intensa e cinematografica: la luce è più drammatica, i dettagli dei tessuti appaiono più realistici e il volto del protagonista è più segnato, suggerendo maggiore stanchezza ed esperienza. Inoltre, i suoi tratti sono più riconducibili a quelli di un uomo italiano degli anni '40, avvicinandosi maggiormente all'iconografia del partigiano. Questo contribuisce a rendere l'immagine più autentica e immersiva, rafforzando l'impatto narrativo della scena.

Figura 4.7: Risultati di generazione dell'immagine di partenza per la scena 3 senza LoRa e con l'utilizzo del modello LoRA creato

4.3 Generazione delle immagini

Dopo aver generato il modello LoRA, si è passati alla definizione di un *workflow* efficiente per la generazione delle immagini per ogni scena. Per la generazione delle immagini è stata utilizzata l'interfaccia ComfyUI, già discussa nel paragrafo 3.2, utilizzando il modello FLUX.1 [dev] descritto nel paragrafo 3.3.2 e il LoRA analizzato nel paragrafo 4.2, creato con Kohya_ss (paragrafo 3.4).

Nella figura 4.8 viene mostrato il *workflow* definitivo e ottimizzato. Dopo aver caricato il modello base FLUX.1 [dev], come già citato, vengono caricati anche i pacchetti aggiuntivi, necessari per il funzionamento del modello, ovvero T5XXL, CLIP e VAE, prima di farli passare attraverso il modello LoRA precedentemente creato. Il nodo che gestisce il modello LoRA consente di controllare l'intensità con cui agiscono il modello base o il pacchetto CLIP rispetto al LoRA stesso, con valori che vanno da zero a uno; questo consente di dare più controllo della generazione al modello personalizzato in alcuni casi in cui la configurazione ordinaria strength_model = 1, strength_clip = 1 non dia risultati soddisfacenti o abbastanza realistici. In alcuni casi la configurazione usata è stata di strength_model = 0.8, strength_clip = 0.7, che ha consentito di avere risultati ottimali.

Il *positive prompt*, ovvero la descrizione testuale che il modello utilizza per generare l'immagine, è stato scritto seguendo delle regole che saranno meglio spiegate nel paragrafo 4.3.1. La lista completa di tutti i prompt utilizzati è invece disponibile nell'appendice B. Dopo aver scritto la descrizione, durante la generazione, il modello CLIP trasforma il testo in un vettore latente, che guida la generazione dell'immagine, influenzandone la composizione, lo stile e il contenuto. Proprio per questo motivo è fondamentale avere un prompt ben scritto e dettagliato, che ottimizzi la coerenza del risultato finale, e proprio per questo motivo sono nati degli studi e sono state sviluppate delle tecniche di *prompt engineering*.

Successivamente, il *sampler* controlla il processo di diffusione dell'immagine, ovvero il processo che converte il rumore latente iniziale nell'immagine finale. In questo caso e in tutti i *workflow* che utilizzano i modelli FLUX.1, il nodo di sampling utilizzato è SamplerCustomAdvanced, che ha una composizione di Input e Widget diversa dai sampler tradizionali:

- Il nodo RandomNoise gestisce il rumore iniziale dell'immagine; per alcuine generazioni è stato impostato casualmente, mentre per altre scene più specifiche sono stati utilizzati dei rumori scelti in precedenza, riportati nella

tabella 4.1, in modo da avere dei risultati sempre abbastanza simili anche cambiando di poco il prompt per adattarlo al tipo di risultato desiderato.

- Una combinazione dei nodi ModelSamplingFlux, BasicScheduler va a finire nell'input `sigmas`, che gestisce la progressione della diffusione dell'immagine; questo valore influenza il grado di dettaglio mantenuto durante il processo di *denoise*.
- Un'ulteriore concatenazione dei ModelSamplingFlux e FluxGuidance con il nodo BasicGuider si inserisce invece nell'input `guider`, che controlla il livello di aderenza alla descrizione testuale; un valore di *guidance* troppo elevato può rendere l'immagine artificiale o eccessivamente influenzata dal testo, mentre un valore troppo basso potrebbe ridurre la fedeltà del risultato desiderato.
- I due nodi width e height si uniscono nel nodo EmptySD3LatentImage prima di andare nell'input `latent_image`; questa struttura crea un'area cosiddetta "latente", un'immagine vuota delle dimensioni specificate, che viene riempita prima col rumore iniziale e progressivamente con il risultato della generazione.
- Infine, il nodo VAE Decode converte il risultato latente finale in un'immagine visibile, riportando l'output in uno spazio RGB comprensibile dall'occhio umano. Dunque, senza il VAE il risultato sarebbe solo una rappresentazione incomprensibile di rumore.

Seed	**Scene**	**Valore**
partigiano	3, 7	184229719894473
lettera	4	716242073457185
consegna lettera	5	143001186633932
paesaggi montagna	6	331831932758698
paesaggi città	1	927715408442632

Tabella 4.1: Lista dei seed utilizzati per la generazione delle immagini

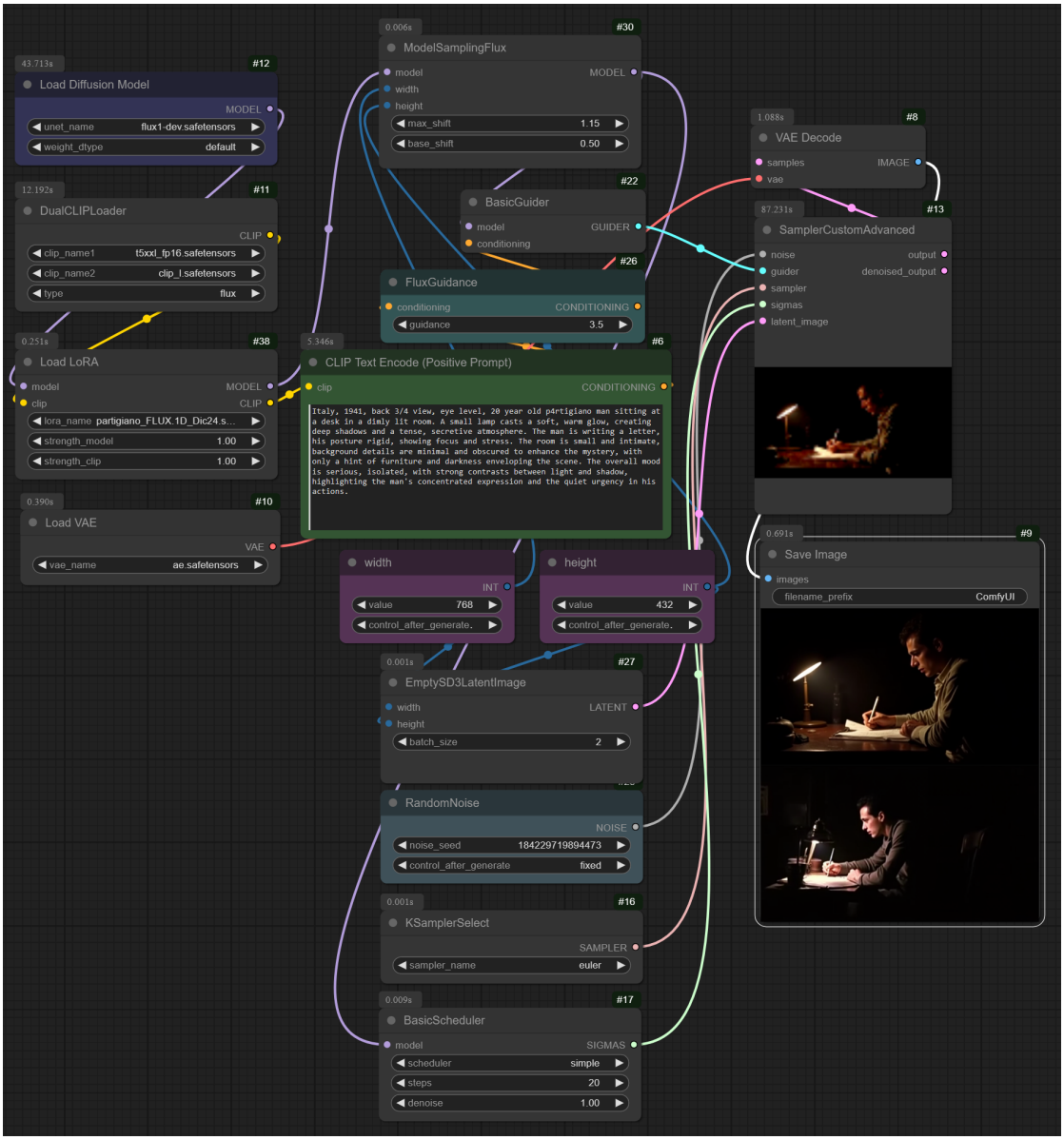


Figura 4.8: *Workflow* di ComfyUI per la generazione delle immagini utilizzando il modello FLUX.1 [dev] e il LoRA personalizzato sul partigiano

4.3.1 Prompt engineering

Con *prompt engineering* si definisce il processo di progettazione e ottimizzazione di richieste testuali (*prompt*), utilizzati per guidare modelli di intelligenza artificiale verso output desiderati, migliorando la loro capacità di adattarsi

e generalizzare a diversi compiti. L'ottimizzazione dei *prompt* implica anche l'esplorazione di vari formati e strutture linguistiche, consentendo di scoprire quali combinazioni producono i risultati più soddisfacenti e appropriati per il contesto narrativo o visivo desiderato [36].

Nel contesto della generazione delle immagini di partenza, da cui sarebbero poi stati creati i video, una regola fondamentale è stata quella di specificare sempre il luogo e l'anno in cui le vicende si svolgevano: tutti i prompt iniziavano sempre con le parole chiave "Italy" e "1939", o l'anno di riferimento per quell'immagine. Inoltre, in ogni immagine in cui era presente il volto del partigiano, è stato necessario inserire la *trigger word* "p4rtigiano man" per richiamare il modello LoRA, come specificato nel paragrafo 4.2.

Le descrizioni sono sempre state composte da singole parole chiave o da brevi frasi separate da virgole e punti. Infatti, l'utilizzo di lunghi periodi o strutture complesse rende difficile la comprensione del testo da parte dei modelli CLIP e del Text Encode.

Alcuni *workflow* includono, inoltre, una seconda descrizione testuale, denominata *negative prompt*: in questo caso, vanno specificate le caratteristiche che non si vogliono vedere nella generazione finale, oltre a eventuali imperfezioni o elementi da evitare. Questo nodo non è presente nei modelli basati su FLUX.1, ma lo si vedrà nel seguito per la generazione dei video.

4.4 Generazione dei video

Dopo aver ottenuto le immagini per ogni scena, il passo successivo è stato trasformare ognuna di esse in una sequenza video che mantenesse coerenza stilistica e narrativa. Anche in questa fase è stata utilizzata l'interfaccia ComfyUI, mentre il modello scelto per la generazione dei video è CogVideoX, già descritto nel paragrafo 3.5, nella sua versione 1.5-5B-I2V.

Nella figura 4.9 viene mostrato il *workflow* utilizzato. Rispetto ai *workflow* per la generazione delle immagini, quelli per i video e soprattutto quelli basati sui modelli CogVideo sono molto rigidi e non consentono di effettuare grosse personalizzazioni. Infatti, in questa fase non è stato possibile inserire il nodo per caricare il LoRA personalizzato sul partigiano: una soluzione è stata richiamarlo all'interno del *prompt* attraverso una funzione apposita fornita da ComfyUI.

Il processo inizia con il caricamento dell'immagine generata nel nodo Load Image, che rappresenta il primo *frame* del video. Talvolta, l'immagine ha subito qualche modifica di post produzione prima di essere caricata per la generazione del video, per correggere o migliorare delle imperfezioni o per eliminare degli artefatti non desiderati. Successivamente, il nodo Resize Image ridimensiona tutte le immagini alla dimensione 768×432 px, utilizzando il metodo di *upscaling* lanczos per mantenere la qualità visiva, e assicura che non vengano elaborati video in qualità troppo alta, per evitare di consumare troppe risorse o incorrere in tempi di generazione estremamente elevati; questo particolare problema non riguarda tanto questo progetto, in cui ci si è accertati in precedenza che le immagini fossero generate alla stessa dimensione citata precedentemente.

Successivamente, dopo aver caricato il modello di CogVideoX, specificando i parametri di precisione e ottimizzazione della memoria per garantire prestazioni adeguate, il sistema utilizza due descrizioni testuali per guidare la generazione del video:

Positive prompt descrive la scena principale del video, riprendendo la stessa descrizione che era stata utilizzata per la generazione dell'immagine di partenza, ma includendo alcuni dettagli sul movimento della camera e dei personaggi.

Negative prompt specifica invece gli errori comuni del modello per escluderli dalla generazione ed evitare che si verifichino, oltre che per migliorare la qualità della generazione. Il prompt negativo utilizzato per tutte le generazioni è "*The video is not of a high quality, it has a low resolution.*

Watermark present in each frame. Strange motion trajectory. Writing or watermark".

Una volta definiti gli input testuali e caricato il modello, si passa alla generazione della sequenza video tramite il nodo CogVideo Sampler, che gestisce i parametri chiave della sintesi video: di fondamentale importanza sono stati i parametri steps e cfg, impostati nella figura 4.9 rispettivamente a 50 e 10.00, ma modificati appositamente per ogni immagine diversa[10]. Il risultato generato dal *sampler* rappresenta una sequenza di immagini in cui il modello ha interpolato i dettagli mancanti e creato il movimento fluido tra i *frame*.

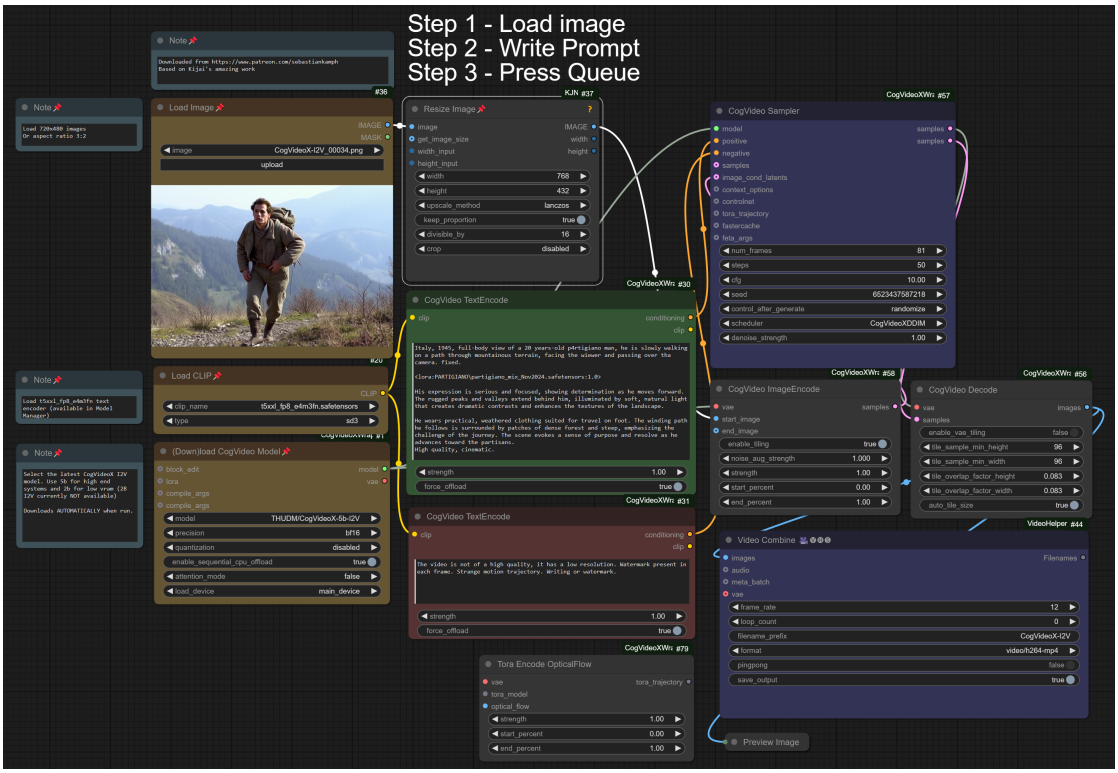


Figura 4.9: *Workflow* di ComfyUI per la generazione dei video utilizzando il modello CogVideoX

[10]Il parametro steps consente di bilanciare la qualità dell'immagine con il tempo di elaborazione. Durante le varie prove si iniziava sempre con il valore predefinito di 25.00 e si incrementava qualora fosse necessario un maggiore livello di dettaglio. Il parametro cfg regola il grado di aderenza del modello alle istruzioni fornite. Valori più elevati di 7.5 determinano un rispetto più rigoroso delle condizioni impostate, risultando particolarmente utili nel caso della generazione video di questo progetto.

Altri nodi di elaborazione dei frame video sono:

CogVideo ImageEncode Converte l'immagine in un formato compatibile con il processo di generazione video di CogVideo, permettendo di integrare le immagini nei pipeline di animazione. Garantisce un'efficiente gestione dei dati visivi per preservare la qualità del video finale.

CogVideo Decode Decodifica i dati video generati in un formato utilizzabile per ulteriori elaborazioni, assicurando che la sequenza mantenga qualità e coerenza visiva prima della fase di esportazione finale.

Tora Encode OpticalFlow Implementa un algoritmo di *optical flow* per migliorare la naturalezza del movimento interpolato, che aiuta a ridurre problemi di transizioni innaturali tra i frame. Questo nodo ha un elevato consumo di risorse, per cui non è stato utilizzato nel *workflow* per contenere i tempi di generazione.

Video Combine Unisce i frame generati e produce il file video finale con le impostazioni di *frame rate* a 12 fps e formato video H.264 mp4.

Nella figura 4.10 viene mostrata una sequenza di frame generati che rappresenta una scena attorno a un falò. In primo piano, due personaggi dialogano, mentre sullo sfondo un altro uomo si avvicina dalle spalle. Il movimento delle fiamme e delle figure contribuisce a rendere la scena dinamica e realistica.

Figura 4.10: Esempio di alcuni frame generati per un video della scena 6

4.5 Upscale in alta qualità e rifinimenti

Come già menzionato nel capitolo precedente, le immagini e i video realizzati finora avevano una qualità abbastanza bassa. La dimensione dei loro canvas era di 768×432 pixel: realizzare i video in alta qualità avrebbe richiesto dei tempi di generazione troppo elevati, oltre a necessitare di risorse che non si avevano a disposizione.

La principale soluzione a questo limite, e una prassi comune nelle *pipeline* degli strumenti di IA Generativa, è quella di affidarsi a degli strumenti di *upscale*. Questi utilizzano modelli basati su reti neurali, come Real-ESRGAN, che consentono di aumentare la risoluzione del video senza introdurre artefatti evidenti. Ogni frame viene elaborato individualmente con un algoritmo di "super-risoluzione", che analizza la struttura dell'immagine e aggiunge dettagli realistici nelle aree poco definite. I principali vantaggi di questa operazione sono una maggiore nitidezza nei dettagli (ad esempio in volti, tessuti e sfondi), la riduzione del rumore generato durante il processo di interpolazione video e una migliore resa delle texture e dei gradienti di colore.

Anche in questo caso, è stato utilizzato un *workflow* di miglioramento sull'interfaccia ComfyUI, mostrato nella figura 4.11 e suddiviso in cinque passi:

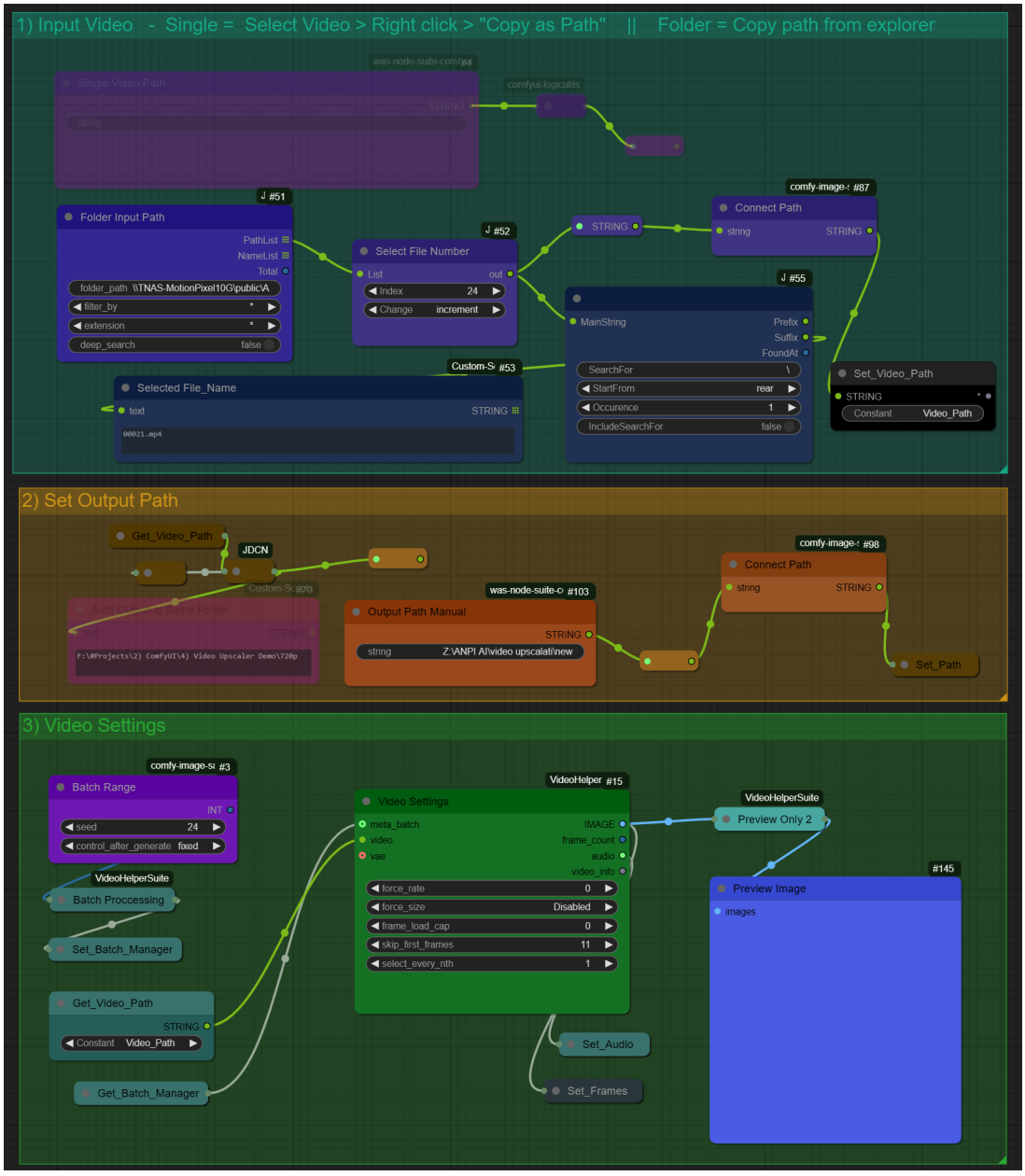
1. Inizialmente vengono selezionati e raccolti tutti i video generati in un'unica cartella, rinominandoli con dei numeri progressivi. Questa fungerà da batch di elaborazione per l'upscale: questo workflow, infatti, integra un sistema che consente di creare una coda di generazione seguendo un indice fornito dall'utente; è necessario inserire le informazioni sul percorso del primo video, fornire una codifica del percorso in cui sono inseriti (se la cartella contiene già tutti i video, basta inserire solo un *back slash*) e i nodi riusciranno a selezionare, per ogni generazione, il corretto video da elaborare. Alternativamente, è possibile anche selezionare un video per volta disattivando questa serie di nodi e attivando quelli relativi alla selezione singola.

2. Il secondo passo consiste nella configurazione del percorso di destinazione. Anche in questo caso ci sono due opzioni: è possibile memorizzare i video prodotti nella stessa cartella di origine, oppure inviarli tutti in un nuovo percorso; nel caso del progetto è stata scelta questa seconda opzione, per poter avere tutti i video migliorati in un'apposita cartella, separata dal resto.

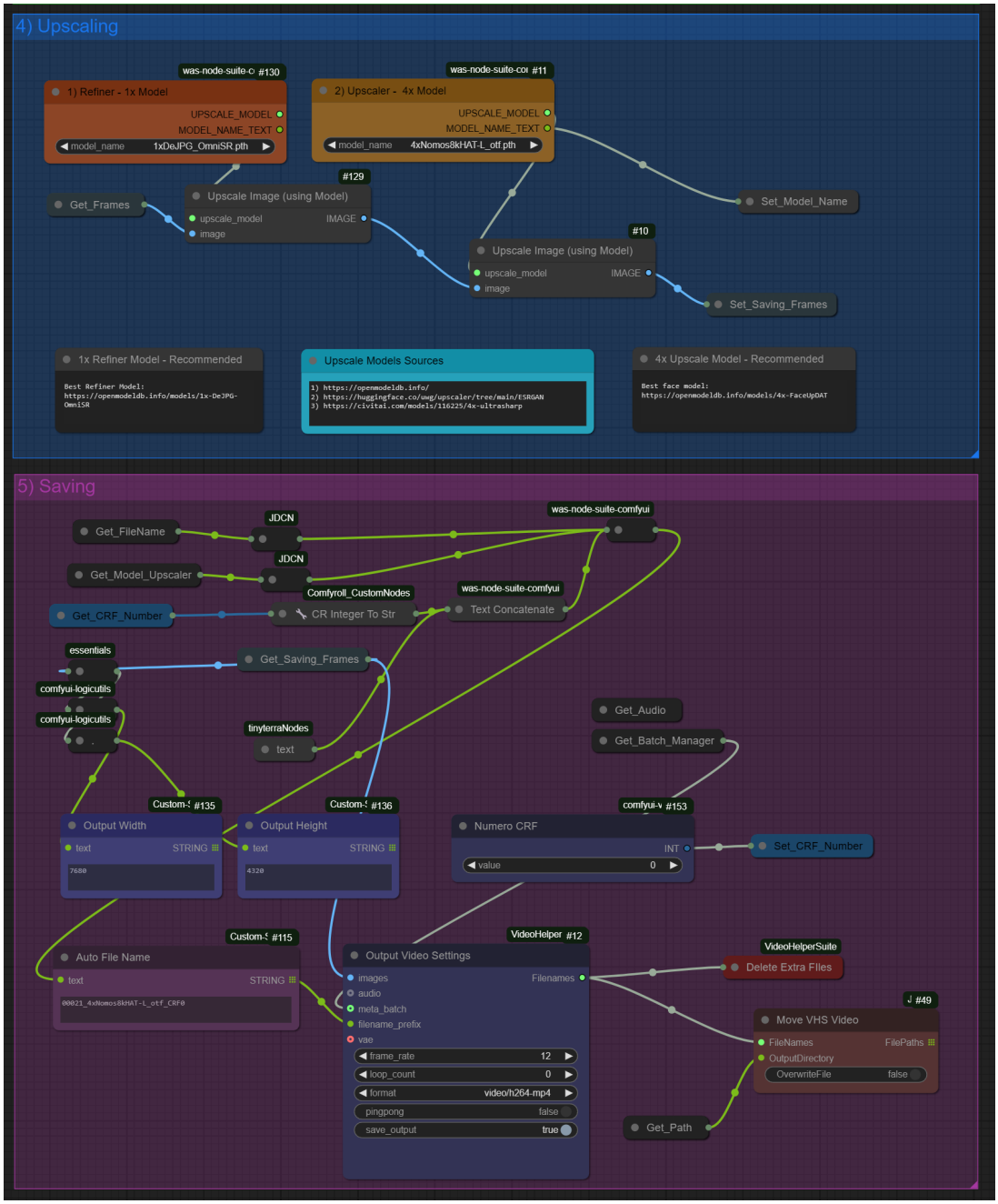
3. Successivamente, si passa alla configurazione delle impostazioni di rendering. In particolare, si consiglia di impostare il Batch Range a 24 per il rendering standard, ma di ridurlo a 5 quando si esegue una conversione da 720p a 4k per evitare problemi di memoria. Inoltre, è sconsigliato ridimensionare forzatamente l'immagine, poiché potrebbe causare artefatti nei pixel durante il processo di *upscale*.

4. Il quarto passo riguarda l'*upscale* vero e proprio. Un primo processo di rifinitura viene effettuato utilizzando il modello 1xDeJPG_OmniSR che incrementa la risoluzione senza aumentare la dimensione dell'immagine: il processo sfrutta tecniche di *deep learning* per ricostruire dettagli mancanti e affinare i bordi, mantenendo la fedeltà visiva e preservando i dettagli essenziali. Successivamente, questo risultato viene inviato al modello di ridimensionamento 4xNomos8kHAT-L_otf che, oltre a effettuare ulteriori rifiniture, aumenta le dimensioni dei frame e riempie i pixel vuoti con informazioni rilevati a ricostruire l'intera immagine in una dimensione quattro volte superiore a quella di partenza, arrivando a una risoluzione finale di 3072×1728 px.

5. Infine, i frame elaborati vengono riaggregati in sequenze video, ricostruendo il file finale con il frame rate originario. Vengono inoltre lette tutte le informazioni riguardanti il processo e inserite nei *metadata* del file video. Il video è salvato nella cartella di destinazione scelta a 12 fps e nel formato video H.264 mp4.



(a) Prima parte del *workflow*



(b) Seconda parte del *workflow*

Figura 4.11: *Workflow* di ComfyUI per l'upscale in alta qualità dei video utilizzando i modelli 1xDeJPG_OmniSR e 4xNomos8k

Dopo il processo di upscale, le sequenze video generate e rifinite sono state montate in un unico trailer, insieme alla colonna sonora e al voice over generati precedentemente, utilizzando il software di montaggio DaVinci Resolve. Il video finale è stato sottoposto a un processo di color grading per uniformare la palette cromatica e correggere eventuali variazioni di tonalità introdotte dai vari modelli di IA Generativa.

Una volta completate anche queste operazioni, il video è stato esportato nel formato standard 1920×1080 px, utilizzando il *codec* H.264, che assicura un'elevata qualità con un file di dimensioni contenute.

Capitolo 5

Conclusioni

L'adozione delle tecnologie di Intelligenza Artificiale Generativa per la produzione audiovisiva rappresenta un campo di ricerca in continua evoluzione. Questo lavoro di tesi ha analizzato le potenzialità e i limiti degli strumenti open source attualmente disponibili, valutando il loro grado di maturità rispetto alle soluzioni proprietarie. L'obiettivo era esplorare fino a che punto sia possibile ottenere contenuti visivamente coerenti e realistici sfruttando esclusivamente modelli aperti, mettendo in evidenza le criticità legate alla generazione di immagini e video.

Dopo un'analisi approfondita dei risultati, emerge come il settore dell'IA Generativa stia vivendo una fase di forte innovazione, ma al contempo presenti ancora diverse sfide tecniche, specialmente per quanto riguarda la consistenza e il controllo del contenuto generato. La differenza tra i modelli open source e quelli proprietari risulta ancora marcata, in particolare nel campo della generazione video, dove la ricerca è meno avanzata rispetto a quella della generazione di immagini.

In questo capitolo vengono discusse le principali osservazioni maturate durante la sperimentazione, suddivise in tre aspetti fondamentali: il realismo e la coerenza dei contenuti generati, i limiti tecnologici riscontrati e le prospettive future dello sviluppo delle IA Generative.

5.1 Realismo e coerenza

Sebbene le immagini generate siano sempre state accurate e abbiano dato risultati qualitativi abbastanza soddisfacenti, a volte, soprattutto per le scene in cui il partigiano assumeva delle posizioni non convenzionali per un modello base di IA generativa (ad esempio, quando era sdraiato nel letto), il modello ha fatto difficoltà a dare dei risultati utilizzabili, storcendo spesso la figura. Il limite principale per la generazione è stato dunque la mancanza di una comprensione strutturale del corpo umano da parte degli attuali modelli di Intelligenza Artificiale.

Il modello FLUX.1 [dev] si è confermato essere la scelta migliore tra i modelli di generazione immagini open source. Forse in questo momento è il modello con la minor varietà e diversificazione dei risultati, ma il suo utilizzo al posto dei modelli basati su Stable Diffusion ha sicuramente garantito un maggior realismo nei contenuti. Inoltre, la sua integrazione con il LoRA personalizzato sulle immagini dei due film sui partigiani ha ulteriormente aumentato questo realismo, migliorando e rendendo più realistiche la grana delle texture, la gestione delle luci e meno netti i bordi degli oggetti.

Per quanto riguarda la generazione dei video, la ricerca dei modelli open source deve ancora arrivare a un livello tale da poter essere considerata allo stato dell'arte. I modelli chiusi e a pagamento analizzati, Suno e Runway *in primis*, grazie anche agli ingenti finanziamenti privati che ricevono, riescono ad avere un'evoluzione molto più rapida rispetto ai modelli aperti e hanno dimostrato un livello qualitativo nettamente superiore. Di conseguenza, mentre l'ecosistema open-source progredisce con una velocità limitata dalle risorse disponibili, le soluzioni commerciali continuano a evolversi rapidamente, consolidando il divario tra le due categorie.

5.2 Limiti

La fase di generazione dei video a partire dalle immagini, in particolare, è stata molto lunga, complessa e ha richiesto un numero elevato di prove per arrivare a un risultato ottimale ma mai perfettamente accettabile. Inoltre, questo processo è stato caratterizzato da un'elevata complessità e da tempi di elaborazione molto lunghi. La necessità di numerose iterazioni per ottenere risultati accettabili ha messo in evidenza i limiti attuali di queste tecnologie, sia in termini di efficienza computazionale che di qualità del risultato.

Un problema particolarmente evidente è stato quello della coerenza temporale. I modelli di IA tendono a generare video in cui ogni frame è trattato come un'immagine indipendente, senza una reale comprensione della continuità del movimento. Ciò ha portato a effetti indesiderati come variazioni involontarie nei volti dei personaggi o fluttuazioni nella forma degli oggetti. Per mitigare questo problema, è stato necessario applicare delle modifiche in post produzione su alcuni elementi e selezionare manualmente i frame migliori.

Un altro limite importante è stato quello relativo alle transizioni tra i frame generati. Spesso, i modelli di IA non riescono a mantenere una coerenza temporale adeguata, producendo artefatti visibili o variazioni incoerenti tra un frame e l'altro. Ciò implica che, allo stato attuale, la generazione video basata su IA non può ancora sostituire completamente le tecniche tradizionali di animazione o ripresa, ma piuttosto rappresenta uno strumento complementare che richiede una fase di rifinitura manuale per ottenere un risultato accettabile.

Infine, è ancora necessario disporre di strumenti hardware potenti e di fascia alta per poter eseguire questi modelli. La generazione di contenuti ad alta risoluzione richiede una notevole quantità di memoria VRAM e potenza computazionale, rendendo l'accesso a queste tecnologie proibitivo per molti creatori indipendenti, soprattutto per quanto riguarda l'ambito open source, che richiede di eseguire tutti i modelli in locale. Questa barriera evidenzia la necessità di una maggiore ottimizzazione e accessibilità per garantire una più ampia adozione di questi strumenti.

5.3 Futuro delle IA Generative

Il 6 marzo 2025 è stata rilasciata la versione di Hunyuan per ComfyUI, introducendo miglioramenti significativi nel flusso di lavoro della generazione video e nell'elaborazione delle immagini. Questo aggiornamento rappresenta un ulteriore passo avanti nella ricerca di strumenti accessibili e open-source capaci di competere con le soluzioni proprietarie. Hunyuan è un modello di Intelligenza Artificiale Generativa per la generazione di video da testo, realizzato dalla più grande azienda tecnologica cinese, la Tencent, già proprietaria di Baidu e WeChat.

Nel frattempo, il settore delle IA generative sta vivendo una fase di consolidamento: gli strumenti più piccoli si stanno unendo tra loro o stanno stringendo collaborazioni commerciali strategiche per evitare di essere eclissati dalle aziende più grandi. Un esempio di questo tipo è quello della collaborazione tra Eleven

Labs e Hedra, due piattaforme chiuse impiegate anche in questo progetto di tesi, che nel luglio 2024 hanno annunciato un accordo per la condivisione e l'integrazione reciproca dei propri servizi. Questo genere di alleanze suggerisce che il mercato tenderà sempre più verso l'aggregazione di pochi attori principali in grado di offrire soluzioni complete e integrate per la generazione automatica di contenuti.

Un altro aspetto chiave sarà l'ulteriore affinamento dell'interazione tra uomo e IA. Strumenti sempre più sofisticati permetteranno ai creatori di mantenere un maggiore controllo artistico sul processo di generazione, superando molte delle limitazioni attuali e avvicinando queste tecnologie a un utilizzo professionale su larga scala. In questo scenario, il ruolo dell'open source sarà determinante per garantire trasparenza, accessibilità e innovazione nel settore.

La speranza più grande per il futuro è che i modelli open source per il video riescano a raggiungere livelli qualitativi soddisfacenti e competitivi rispetto ai modelli proprietari. Un esempio promettente in questa direzione è il modello Text-To-Video sviluppato da FLUX.1, che potrebbe rappresentare un significativo passo avanti per il settore. Tuttavia, al momento della pubblicazione di questa tesi, non vi sono informazioni certe né sullo stato di avanzamento del progetto, né su una sua eventuale pubblicazione imminente.

Appendice A

Sceneggiatura del trailer

SEZIONE 1: L'ITALIA DEVASTATA DALLA GUERRA

SCENA 1: EST. GIORNO – TORINO

Panoramica di Torino semi-deserta. Le strade sono desolate, macerie ovunque, alcuni edifici distrutti. Militari tedeschi pattugliano in lontananza, camminando tra le ombre degli edifici.

PARTIGIANO (V.O.)
In quegli anni c'era la guerra, c'erano i tedeschi, ci si arrangiava come si poteva e chi riusciva scappava dalla città.

SCENA 2: EST. NOTTE – TORINO

Torino bombardata durante la notte. Esplosioni in lontananza. Una fioca luce emerge dalla finestra di una casa.

PARTIGIANO (V.O.)
Torino è stata bombardata. E non era mica bello. nè! Le bombe fischiavano.

SEZIONE 2: NASCONO I PARTIGIANI – LA LETTERA

SCENA 3: INT. NOTTE – CASA PARTIGIANO

Il PARTIGIANO si rigira insonne nel letto, lo sguardo fisso davanti a sé.

PARTIGIANO (V.O.)
Io ero tormentato, volevo fare qualcosa, avevo paura ma sentivo che dovevo muovermi...

SCENA 4: INT. NOTTE – CASA PARTIGIANO

Inquadratura da dietro a tre quarti: il partigiano è seduto alla scrivania, in una stanza cupa illuminata da un piccolo lumino, e sta scrivendo una lettera. Il clima è molto teso e c'è molta segretezza, si vede poco del contesto in cui è.

PARTIGIANO (V.O.)
Fuori non c'erano i bombardamenti ma c'erano tedeschi dappertutto... I fascisti... I primi partigiani andavano in montagna per organizzarsi e stare al sicuro.

Una penna scrive una lettera su un foglio.

SCENA 5: EST. GIORNO – MONTAGNA

Il partigiano attraversa dei paesaggi montani diretto verso i partigiani. Arrivato al rifugio dei partigiani, consegna la lettera a un generale. I due si scambiano uno sguardo rapido e deciso.

PARTIGIANO (V.O.)
Dopo i primi contatti decisi anch'io di fare il mio.

SEZIONE 3: LA RESISTENZA, L'INVERNO

SCENA 6: EST. GIORNO – MONTAGNA E CITTÀ

Si susseguono vari immagini di battaglie in montagna. Un partigiano nascosto, avvolto da un cappotto consumato e trema dal freddo. Al centro di una piazza in città, una persona è impiccata. Non si vede molto della figura.

PARTIGIANO (V.O.)
Furono anni durissimi, l'inverno del '44 fu il peggiore, la vita in montagna era tosta...

SEZIONE 4: IL PARTIGIANO LIBERO OGGI

SCENA 7: INT. GIORNO – SALOTTO DI UNA CASA

Da mezzo busto a primo piano: il partigiano, ormai anziano, guarda in camera e si commuove mentre parla.

PARTIGIANO
Questi sono i miei ricordi, questa è la mia storia. Ve la racconto come viene, mescolando avvenimenti e memoria, perché tutti i ricordi non si perdano come lacrime nella pioggia.

SCENA 8: CARTELLO FINALE

TESTO SU SFONDO NERO: "Dedicato alla vite di tutti partigiani, che siano anche la nostra memoria."

Appendice B

Promptlist

SEZIONE 1: L'ITALIA DEVASTATA DALLA GUERRA

SCENA 1: EST. GIORNO - TORINO

Prompt:
Italy, 1944, a wide, desolate view of Turin during WWII. The streets are empty and strewn with rubble, with several buildings partially destroyed, their facades crumbling and windows shattered. In the distance, small groups of German soldiers patrol cautiously, their dark silhouettes moving through the shadows cast by the standing structures. The light is muted and overcast, casting a somber tone over the desolation. Smoke rises faintly from distant ruins emphasizing the devastation. The atmosphere is heavy and tense, capturing the oppressive quiet of a city under occupation. Cinematic, high detail, realistic.

SCENA 2: EST. NOTTE - TORINO

Prompt:
Italy, 1944, a nighttime view of bombed Turin during WWII. The city is cloaked in darkness, with distant explosions lighting up the skyline, casting brief, sharp glows on the ruins. A faint, warm light escapes from the window of a damaged house, hinting at life

amid the devastation. The streets are strewn with rubble, and the shadows of crumbling buildings loom over the scene. Smoke drifts lazily through the cold night air, adding to the oppressive silence between blasts. Cinematic, high detail, dramatic lighting.

SEZIONE 2: NASCONO I PARTIGIANI - LA LETTERA

SCENA 3: INT. NOTTE - CASA PARTIGIANO

Prompt:
Italy, 1941, 25 year old p4rtigiano man, in a white night dress, overhead perspective, sleepless, his hands are behind his head, expressing worry. The room is dimly lit by the moonlight coming from the window, casting soft and cold light, creating cool night shadows. Focus on realism in the textures, especially the fabric of the nightdress and the worn bed linens. The scene conveys the p4rtigiano man's anxiety and inner turmoil, tied to his role as a partisan in Italy during WWII. High detail, realism, cinematography, masterpiece.

SCENA 4: INT. NOTTE - CASA PARTIGIANO

Prompt:
Italy, 1941, back 3/4 view, eye level, 25 year old p4rtigiano man sitting at a desk in a dimly lit room. A small lamp casts a soft, warm glow, creating deep shadows and a tense, secretive atmosphere. The man is writing a letter, his posture rigid, showing focus and stress. Background details are minimal and obscured to enhance the mystery, with only a hint of furniture and darkness enveloping the scene. The overall mood is serious, isolated, with strong contrasts between light and shadow, highlighting the man's concentrated expression and the quiet urgency in his actions.

SCENA 5: EST. GIORNO - MONTAGNA

Prompt:
Italy, 1945, full-body view of a 25 years-old

p4rtigiano man, walking through mountainous terrain, facing the viewer. His expression is serious and focused, showing determination as he moves forward. The rugged peaks and valleys extend behind him, illuminated by soft, natural light that creates dramatic contrasts and enhances the textures of the landscape. He wears practical, weathered clothing suited for travel on foot. The winding path he follows is surrounded by patches of dense forest and steep, rocky slopes, emphasizing the challenge of the journey. The scene evokes a sense of purpose and resolve as he advances toward the partisans.

SEZIONE 3: LA RESISTENZA, L'INVERNO

SCENA 6: EST. GIORNO – MONTAGNA E CITTÀ

Prompt:
Italy, 1939, an expansive view of a rural dirt road cutting through vast fields and sparse trees, with 4 German soldiers setting up a roadblock in the distance. An old car approaches slowly along the road, its headlights faintly visible. Farther back, a small village with rustic houses and a church tower is nestled against rolling hills. Cinematic, high detail, realistic, masterpiece.

Italy, 1939, a winding dirt road at the foot of the rolling hills in the valleys of Piedmont, bordered by fields and sparse trees. A small group of German soldiers mans a roadblock, their distinct uniforms and stern expressions adding to the tension. Rifles are slung over their shoulders as they inspect the area. An old car approaches cautiously along the road, its silhouette outlined against the backdrop of the hills. In the distance, a small village nestles into the base of the hills. The natural, diffused light of the valley casts soft shadows, enhancing the quiet yet foreboding atmosphere. Cinematic, high detail, realistic, masterpiece, colors.

Italy, 1944, a quiet, small Italian village under fascist surveillance during WWII. In foreground a

couple of fascist soldiers in military coats patrols the area, their presence imposing and authoritative, rifles slung over their shoulders. The buildings are traditional, with weathered facades and red-tiled roofs, creating a distinctly rural Italian setting. A small military van is at the end of the road. The lighting is subdued and natural, casting soft shadows that emphasize the tense atmosphere. The scene captures the oppressive mood and the constant watchfulness of wartime occupation. Cinematic, high detail, realistic.

SEZIONE 4: IL PARTIGIANO LIBERO OGGI

SCENA 7: INT. GIORNO – SALOTTO DI UNA CASA

Prompt:
Italy, close-up of an elderly p4rtigiano man speaking directly to the camera, his face weathered and lined with the marks of age and hardship. His eyes glisten with emotion, reflecting deep, poignant memories as he shares his wartime experiences. A tear forms and slowly rolls down his cheek, adding to the rawness of the moment. The lighting is soft and warm, highlighting the details of his features and conveying a sense of intimacy and vulnerability. The background is simple and blurred, focusing all attention on his expressive face. The scene is filled with nostalgia, sorrow, and a quiet strength as he recounts his story.

Bibliografia

[1] Leonardo Banh e Gero Strobel. «Generative artificial intelligence». In: *Electronic Markets* 33.1 (2023), p. 63. URL: https://doi.org/10.1007/s12525-023-00680-1 (cit. a p. 1).

[2] Aditi Singh. «A Survey of AI Text-to-Image and AI Text-to-Video Generators». In: *2023 4th International Conference on Artificial Intelligence, Robotics and Control (AIRC)*. 2023, pp. 32–36. DOI: 10.1109/AIRC57904.2023.10303174 (cit. a p. 2).

[3] Rui Sun, Yumin Zhang, Tejal Shah, Jiahao Sun, Shuoying Zhang, Wenqi Li, Haoran Duan, Bo Wei e Rajiv Ranjan. «From Sora What We Can See: A Survey of Text-to-Video Generation». In: *ArXiv* abs/2405.10674 (2024). URL: https://doi.org/10.4018/IJSSCI.300364 (cit. a p. 2).

[4] Zhuoyi Yang et al. *CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer*. 2024. arXiv: 2408.06072 [cs.CV]. URL: https://arxiv.org/abs/2408.06072 (cit. alle pp. 3, 33).

[5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser e Björn Ommer. «High-resolution image synthesis with latent diffusion models». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695 (cit. alle pp. 4, 23, 24).

[6] OpenAI. *DALL·E 3 System Card*. [Online]. Ott. 2023. URL: https://openai.com/index/dall-e-3-system-card/ (visitato il giorno 07/02/2025) (cit. a p. 4).

[7] Midjourney. *Midjourney Documentation*. [Online]. 2025. URL: https://docs.midjourney.com/ (visitato il giorno 07/02/2025) (cit. a p. 5).

[8] OpenAI. *Sora System Card*. [Online]. Dic. 2024. URL: https://openai.com/index/sora-system-card/ (visitato il giorno 07/02/2025) (cit. alle pp. 5, 6).

[9] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog e Anastasis Germanidis. «Structure and content-guided video synthesis with diffusion models». In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 7346–7356. arXiv: 2302.03011 [cs.CV]. URL: https://arxiv.org/abs/2302.03011 (cit. a p. 5).

[10] Runway. *Gen-3 Alpha Documentation*. [Online]. 2025. URL: https://help.runwayml.com/hc/en-us/sections/30265301423635-Gen-3-Alpha (visitato il giorno 07/02/2025) (cit. a p. 6).

[11] OpenAI. *Video Generation Models as World Simulators*. [Online]. 2024. URL: https://openai.com/index/video-generation-models-as-world-simulators/ (visitato il giorno 08/02/2025) (cit. a p. 6).

[12] Francisco Eiras et al. *Risks and Opportunities of Open-Source Generative AI*. 2024. arXiv: 2405.08597 [cs.LG]. URL: https://arxiv.org/abs/2405.08597 (cit. a p. 7).

[13] Ingrid Campo-Ruiz. «Artificial intelligence may affect diversity: architecture and cultural context reflected through ChatGPT, Midjourney, and Google Maps». In: *Humanities and Social Sciences Communications* 12.1 (2025), pp. 1–13. URL: https://doi.org/10.1057/s41599-024-03968-5 (cit. a p. 8).

[14] Film Commission Torino Piemonte. *Motion Pixel - Scheda di produzione*. [Online]. 2025. URL: https://www.fctp.it/production_item.php?id=2808 (visitato il giorno 09/02/2025) (cit. a p. 10).

[15] Stefano Conca Bonizzoni. *Sweet End of the World!* [Online]. Documentario, 16 minuti. 2024. URL: https://www.cinemaitaliano.info/film/34094/festival/sweet-end-of-the-world.html (visitato il giorno 12/02/2025) (cit. a p. 10).

[16] ANPI Chiomonte Alta Valle Susa. *Sito ufficiale ANPI Chiomonte Alta Valle Susa*. [Online]. 2025. URL: https://www.anpichiomonteavs.it/ (visitato il giorno 09/02/2025) (cit. a p. 11).

[17] La Stampa. *Morto a 96 anni Cesare Alvazzi del Frate, comandante partigiano*. [Online]. 2023. URL: http://lastampa.it/torino/2023/01/18/news/morto_96_anni_cesare_alvazzi_del_frate_resistenza_alta_val_di_susa-12590051 (visitato il giorno 09/02/2025) (cit. a p. 12).

[18] ComfyUI Documentation Team. *Core Concepts - Workflow*. [Online]. 2025. URL: https://docs.comfy.org/essentials/core-concepts/workflow (visitato il giorno 13/02/2025) (cit. a p. 19).

[19] ComfyUI Documentation Team. *Core Concepts - Properties*. [Online]. 2025. URL: https://docs.comfy.org/essentials/core-concepts/properties (visitato il giorno 18/02/2025) (cit. a p. 21).

[20] ComfyUI Community. *KSampler - ComfyUI Community Manual*. [Online]. 2025. URL: https://blenderneko.github.io/ComfyUI-docs/Core%20Nodes/Sampling/KSampler/#inputs (visitato il giorno 25/02/2025) (cit. a p. 22).

[21] Patrick Esser et al. «Scaling rectified flow transformers for high-resolution image synthesis». In: *Forty-first international conference on machine learning*. 2024 (cit. a p. 23).

[22] Stability AI. *Partners*. [Online]. 2025. URL: https://stability.ai/partners (visitato il giorno 25/02/2025) (cit. a p. 23).

[23] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna e Robin Rombach. *SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis*. Lug. 2023. arXiv: 2307.01952 [cs.CV] (cit. a p. 25).

[24] Black Forest Labs. *Announcing Black Forest Labs - Black Forest Labs*. [Online]. Ago. 2024. URL: https://blackforestlabs.ai/announcing-black-forest-labs/ (visitato il giorno 01/03/2025) (cit. a p. 26).

[25] Stable Diffusion Art. *SDXL vs FLUX - Which Model Generates Better Images?* [Online]. 2024. URL: https://stable-diffusion-art.com/sdxl-vs-flux/ (visitato il giorno 04/03/2025) (cit. a p. 28).

[26] Black Forest Labs. *Up Next - Black Forest Labs*. [Online]. 2025. URL: https://blackforestlabs.ai/up-next/ (visitato il giorno 01/03/2025) (cit. a p. 29).

[27] bmaltais. *kohya_ss - Kohya's GUI*. [Online]. 2025. URL: https://github.com/bmaltais/kohya_ss?tab=readme-ov-file#kohyas-gui (visitato il giorno 03/03/2025) (cit. a p. 30).

[28] Reece Shuttleworth, Jacob Andreas, Antonio Torralba e Pratyusha Sharma. *LoRA vs Full Fine-tuning: An Illusion of Equivalence*. 2024. arXiv: 2410.21228 [cs.LG] (cit. a p. 30).

[29] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu e Jie Tang. *CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers*. 2022. arXiv: 2205.15868 [cs.CV] (cit. a p. 34).

[30] ElevenLabs. *Voice Conversion: Making one person speak in the voice of another*. [Online]. Set. 2022. URL: https://elevenlabs.io/blog/voice-conversion (visitato il giorno 06/03/2025) (cit. a p. 35).

[31] ElevenLabs. *Introducing Eleven Multilingual v1: Our New Speech Synthesis Model*. [Online]. Apr. 2023. URL: https://elevenlabs.io/blog/eleven-multilingual-v1 (visitato il giorno 06/03/2025) (cit. a p. 35).

[32] ElevenLabs. *This Voice Doesn't Exist - Generative Voice AI*. [Online]. Gen. 2023. URL: https://elevenlabs.io/blog/enter-the-new-year-with-a-bang (visitato il giorno 06/03/2025) (cit. a p. 35).

[33] The Generator. «How I made AI sing: Using Suno and Hedra to generate niche indie pop and emo polka songs». In: *Medium - The Generator* (gen. 2025). [Online]. URL: https://medium.com/the-generator/how-i-made-ai-sing-using-suno-and-hedra-to-generate-niche-indie-pop-and-emo-polka-songs-1a998d25cea6 (visitato il giorno 08/03/2025) (cit. a p. 36).

[34] Jiaxing Yu, Songruoyao Wu, Guanting Lu, Zijin Li, Li Zhou e Kejun Zhang. «Suno: potential, prospects, and trends». In: *Frontiers of Information Technology & Electronic Engineering* 25.7 (2024), pp. 1025–1030. DOI: 10.1631/FITEE.2400299 (cit. a p. 37).

[35] Laura Carnevali. «How to Fine-Tuning a Model with Kohya». In: *Intelligent Art* (giu. 2024). [Online]. URL: https://medium.com/intelligent-art/how-to-fine-tuning-a-model-with-kohya-e05922dc1033 (visitato il giorno 20/03/2025) (cit. a p. 45).

[36] Jiaqi Wang et al. «Review of large vision models and visual prompt engineering». In: *Meta-Radiology* 1.3 (2023), p. 100047. ISSN: 2950-1628. DOI: https://doi.org/10.1016/j.metrad.2023.100047 (cit. a p. 52).