UNIVERSITÀ DEGLI STUDI DI UDINE
HIC SUNT FUTURA

DIPARTIMENTO DI SCIENZE MATEMATICHE, INFORMATICHE E FISICHE

TESI DI LAUREA MAGISTRALE IN INFORMATICA

IA per il retrieval di esibizioni d'arte multimediale per il Metaverso

CANDIDATO

Gianluca Macrì

RELATORE

Prof. Giuseppe Serra

CORRELATORE

Ph.D. Alex Falcon

Anno accademico 2023-2024

Contatti dell'Istituto
Dipartimento di Scienze Matematiche, Informatiche e Fisiche
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine — Italia
+39 0432 558400
https://www.dmif.uniud.it/

Indice

Introduzione

Negli ultimi anni, si è sempre più spesso sentito parlare di metaverso. Tipicamente, ciò avviene in relazione agli sviluppi tecnologici in termini di visori di realtà virtuale e come settore oggetto di ingenti investimenti da parte di colossi tecnologici come Microsoft e Meta. Più precisamente, con il termine "metaverso", introdotto per la prima volta nel 1997 dallo scrittore Neal Stephenson nel romanzo di fantascienza "Snow crash", ci si riferisce oggi, secondo la definizione dell'Osservatorio Extended Reality & Metaverse, ad un "ecosistema immersivo, persistente, interattivo e interoperabile, composto da molteplici mondi virtuali interconnessi in cui gli utenti possono socializzare, lavorare, effettuare transazioni, giocare e creare asset, accedendo anche tramite dispositivi immersivi" [47].

Un tale ambiente si presta dunque alla realizzazione di diverse applicazioni, spaziando dall'ambito lavorativo, tramite la creazione di sale riunioni virtuali [70] e procedure di addestramento in ambito chirurgico [45], a quello più legato al mondo dello svago, potendo ospitare social network particolarmente immersivi, come ad esempio Decentraland[1], o eventi culturali.

Riguardo a quest'ultima categoria, e guardando in particolar modo al mondo dell'arte, già ad oggi esistono alcune fiere virtuali che coinvolgono artisti internazionali. Per esempio, è il caso della Art Week[2] o della metaverse art fair[3], oltre ad alcune gallerie d'arte virtuali [21, 10].

Seppur nel complesso tali mondi risultino ad oggi ancora in numero limitato, la previsione del crescente numero di utenti [46] e della conseguente richiesta di contenuti e spazi virtuali potrebbe rapidamente portare ad un massiccio incremento di tali ambienti, facendo emergere il problema della ricerca degli elementi più interessanti e rilevanti da parte degli utenti, analogamente a quanto accaduto con la rapida diffusione di contenuti informativi sul web. A testimonianza di questa tendenza, ed in particolare proprio nell'ambito delle applicazioni in ambito culturale, già nel 2020 il ministero della cultura e del turismo cinese riportava di come durante l'ultimo capodanno lunare i musei della regione avessero realizzato e offerto in rete oltre 2000 diverse esperienze di visite virtuali[26].

Diventa dunque fondamentale poter disporre di sistemi simili agli odierni motori di ricerca che consentano di navigare il cyberspazio in modo efficiente, ad esempio attraverso l'utilizzo di query testuali [4].

[1] `https://decentraland.org/`
[2] https://decentraland.org/artweek/
[3] https://m-art.io/

La ricerca di metaversi a partire da una query testuale fornita dall'utente affonda le sue radici nella più attiva area di ricerca denominata Information Retrieval, nata nel contesto della ricerca di informazioni all'interno di grandi collezioni di documenti e sviluppatasi in sinergia col mondo dell'Intelligenza Artificiale. In particolare, ciò è avvenuto con la necessità di gestire e organizzare grandi collezioni di dati multimediali, quali ad esempio immagini e video, ottenendo ottimi risultati [39, 42]. Più di recente, proprio in considerazione alle previsioni di crescita legate al metaverso, sta nascendo una nuova sotto-area di ricerca legata al retrieval di scenari tridimensionali complessi [2, 19] denominata metaverse-retrieval.

Nell'ambito di questo lavoro di tesi si propone di indagare e contribuire agli sviluppi iniziali di tale area, analizzando più nel concreto uno scenario legato al mondo delle esposizioni d'arte. Tale applicazione dell'utilizzo di spazi virtuali, analogamente a quella delle esposizioni museali nel metaverso [19], presenta la peculiarità di poter contenere degli elementi multimediali di vario genere, come ad esempio video informativi utili a guidare il visitatore nella visita dell'esposizione oppure esempi di video intesi come vere e proprie opere d'arte, che li rendono particolarmente interessanti e sfidanti per l'applicazione di tecniche di retrieval. Infatti, ci sono due principali sfide nell'analisi di questi dati: l'eterogeneità degli elementi multimediali presenti all'interno del museo, quali ad esempio quadri, video, sculture ed artefatti, ed il fatto che ognuno di questi elementi multimediali influenza la pertinenza di uno specifico museo agli occhi dell'utente. Ciò comporta la necessità di implementare un approccio che analizzi le molteplici modalità sia separatamente, rispettando le loro peculiarità, sia congiuntamente, per coglierne le relazioni e le connessioni trasversali.

I contributi di questo lavoro di tesi si possono suddividere in due gruppi distinti legati rispettivamente alla creazione di un dataset inerente allo scenario preso in considerazione e all'applicazione di modelli di intelligenza artificiale per affrontare il problema del retrieval automatico su di essi. Per quando riguarda il dataset, il primo passo ha riguardato la realizzazione di un dataset di opere d'arte video, mentre la loro integrazione, assieme a quadri ed altre opere artistiche, ha portato alla successiva realizzazione di un dataset di esibizioni artistiche virtuali. In particolare, in questo primo contributo, sono stati sviluppati una serie di programmi per la generazione automatica di tali ambienti virtuali, rappresentati come scene tridimensionali, integrandovi all'interno le opere artistiche. Rispetto al problema del retrieval automatico, si sono allenati e testati alcuni modelli noti sul dataset di video artistici, e si è andati a proporre ed indagare l'utilizzo di nuove architetture nel caso del più complesso task di metaverse-retrieval.

Il presente lavoro di tesi viene organizzato secondo la seguente struttura. Nel primo capitolo si affronteranno gli aspetti legati ad alcune nozioni teoriche utili alla comprensione del seguito, introducendo il problema del retrieval automatico nello specifico caso di retrieval inter-modale e introducendo le principali architetture e dataset utilizzati in questo ambito. Nel capitolo 2 "Metodologia e implementazione" verranno descritte congiuntamente la metodologia seguita e alcuni dei dettagli implementativi, suddividendo la trattazione tra la creazione dei dataset e la scelta ed

organizzazione delle architetture di Intelligenza Artificiale. Il terzo capitolo riguarderà invece lo svolgimento di diversi esperimenti di allenamento dei modelli definiti nel capitolo precedente e la discussione critica dei risultati ottenuti. La tesi concluderà la trattazione del lavoro con un breve capitolo finale che ripercorre i punti principali di esso, mettendone in evidenza le limitazioni e i possibili sviluppi futuri. Infine si aggiungono due appendici A e B, dedicate rispettivamente a fornire alcuni dettagli aggiuntivi sugli elementi dei dataset e a rispondere ad eventuali dubbi di natura più tecnica riguardo all'implementazione.

1

Background teorico e lavori correlati

In questo capitolo si darà una visione generale su quello che è il panorama della ricerca in relazione all'abito di questo lavoro di tesi, fornendo gli elementi base per la comprensione dei capitoli successivi. In particolare, dopo aver inquadrato e definito il problema del retrieval automatico, ne si tratteranno le declinazioni in relazione a dati di vario tipo: immagini, video e scenari 3D, introducendo le architetture più comunemente utilizzate, base dell'implementazione descritta nel capitolo 2. Nell'ultima parte si introdurranno brevemente le diverse tipologie di dataset pubblicamente disponibili ed impiegati in questo ambito.

1.1 Retrieval automatico

In un mondo sempre più interconnesso e ricco di informazioni il campo dell'Information Retrieval si occupa di sviluppare tecniche per la loro organizzazione e recupero, cercando di ridurre al minimo il divario con l'utente finale. Se tradizionalmente tale disciplina si concentrava su documenti testuali, lo sviluppo delle capacità della rete e l'avvento dei social media hanno portato ad una diffusione massiccia di contenuti multimediali, creando nuove sfide ed opportunità di sviluppo per tale settore. In particolare una delle problematiche chiave da affrontare consiste nel riuscire ad estrarre informazioni utili da tali tipologie di dati, al fine di poterli mettere in relazione gli uni con gli altri, permettendo, ad esempio, agli utenti di eseguire delle ricerche (query) testuali per recuperare delle immagini o viceversa.

Si parla in quest'ultimo caso di retrieval inter-modale (cross-modal retrieval), indicando appunto la possibilità di recuperate elementi appartenenti ad una certa tipologia di dato a partire da una query di altro genere. In questo contesto l'intelligenza artificiale, ed in particolare i modelli di deep learning, hanno giocato un ruolo chiave grazie alla loro capacità di estrazione di rappresentazioni semantiche astratte a partire da dati con tipologie molto eterogenee.

Di conseguenza, anche il campo dell'information retrieval, come molti altri, è stato influenzato dal successo del Transformer, divenuto celebre in seguito alla pubblicazione del paper "Attention is all you need"[72]. Tale architettura infatti, sebbene inizialmente proposta nell'ambito del na-

tural language processing, è poi stata applicata, grazie alla sua versatilità, a dati di ogni tipologia (immagini[17, 57] , video [42, 41, 31] , audio[25, 30] ...), secondo una filosofia riassumibile col motto "anything you can tokenize, you can feed to Transformer" [11].

1.1.1 Retrieval di immagini con modelli basati su CLIP

Nell'ambito del retrieval cross-modal, ed in particolare considerando il caso del retrieval text-to-image un modello di grande successo è sicuramente CLIP ("Contrastive Language-Image Pre-training") [57], proposto dai ricercatori di OpenAI all'inizio del 2021. L'intento fondamentale degli autori era quello di superare le limitazioni intrinseche dei sistemi di computer vision allora stato dell'arte nell'ambito della classificazione, basati su un numero predefinito di categorie. L'intuizione fu quella di andare piuttosto a sfruttare grandi quantità di dati per poter associare direttamente immagini e rappresentazioni testuali in linguaggio naturale, così da ottenere un sistema molto più flessibile.

In particolare, l'architettura di base prevede un approccio di tipo dual-stream [80, 38] in cui gli elementi appartenenti alle due diverse modalità, in questo caso testuale e visiva, vengono elaborati in modo indipendente mediante due codificatori (encoder) distinti, così da poterne sfruttare le peculiarità . Le rappresentazioni così ottenute possono quindi essere allineate in uno spazio multidimensionale comune, dove le rappresentazioni di elementi semanticamente simili appaiano vicine le une alle altre e sufficientemente separate da quelle di elementi dissimili, indipendentemente dalla modalità.

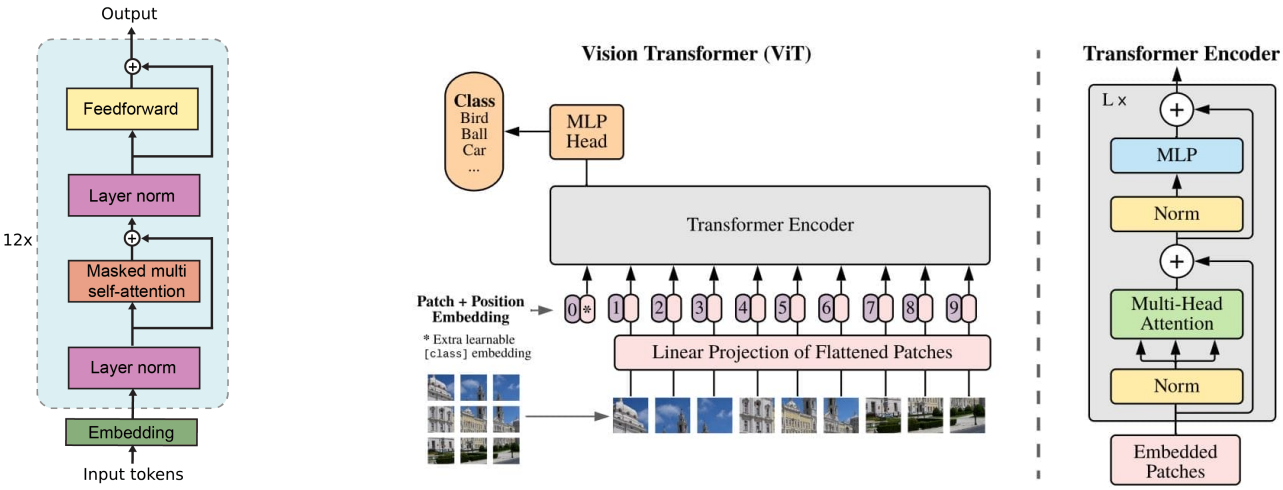
Nello specifico, gli autori hanno optato per l'utilizzo di due encoder basati entrambi sull'architettura del transformer: una versione modificata del transformer originale per l'encoder testuale e il Vision Transformer[17] per il caso visivo. Come si vedrà anche nelle sezioni successive, tali modelli, brevemente descritti in seguito, costituiranno gli elementi base anche per il caso del retrieval di video.

Encoder testuale

Per quanto riguarda l'encoder testuale, in particolare, è stata utilizzata la versione del decoder introdotta per la famiglia di modelli GPT-2 [58]. Tale architettura, rappresentata schematicamente in figura 1.1a, è costituita dagli elementi caratteristici del transformer, utilizzando però una pre-normalizzazione delle feature all'interno dei blocchi di computazione al fine di ottenere una maggiore stabilità dei modelli.

Analizzando tale struttura, si può subito riconoscere la suddivisione interna delle unità nelle due operazioni distinte di applicazione del meccanismo della multi-head self-attention e di elaborazione successiva attraverso dei layer lineari. Tali operazioni di trasformazione possono essere matematicamente formulate come

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^0 \tag{1.1}$$



(a) Architettura dei modelli GPT-2 [9]

(b) Architettura del Vision Transformer[17]

Figura 1.1: Strutture degli encoder testuale e visuale utilizzati per CLIP[57]

dove $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ e $\text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_h}}\right) V_i$, secondo l'usuale suddivisione dell'input in vettori key, query e value, e

$$\text{FeedForward}(x) = \text{ReLU}\left(xW_1 + b_1\right) W_2 + b_2$$

L'operazione di multi-head self-attention, in particolare, costituisce uno degli ingredienti caratteristici delle architetture del transformer e di quelle basate su di essa. Per ciascuna "testa", infatti, l'utilizzo di tale computazione permette di confrontare ciascun elemento della sequenza di input (query) con tutti gli altri elementi (key) derivandone una misura di similarità. Tali valori possono quindi essere utilizzati, per ciascuna query, al fine di sintetizzare una nuova rappresentazione derivata dalla somma pesata degli altri elementi (value), e combinata con quella iniziale, in modo da arricchirne la semantica.

Encoder visivo

L'encoder visivo utilizza in CLIP[57] è invece basato sull'architettura del Vision Transformer [17], che riprende nuovamente l'architettura base del transformer, applicandola però a dati di tipo visivo. In questo caso, per poter convertire un'immagine in una sequenza di token da poter elaborare mediante i blocchi del modello, gli autori hanno seguito un approccio basato su patch, suddividendo ciascun immagine in porzioni da 16×16 pixel per creare dei token. Questi elementi vengono quindi convertiti, attraverso un'operazione di proiezione, in una rappresentazione vettoriale monodimensionale che, aumentata dall'informazione posizionale, può essere elaborata come rappresentato nella figura 1.1b per estrarre una feature finale dell'immagine di partenza.

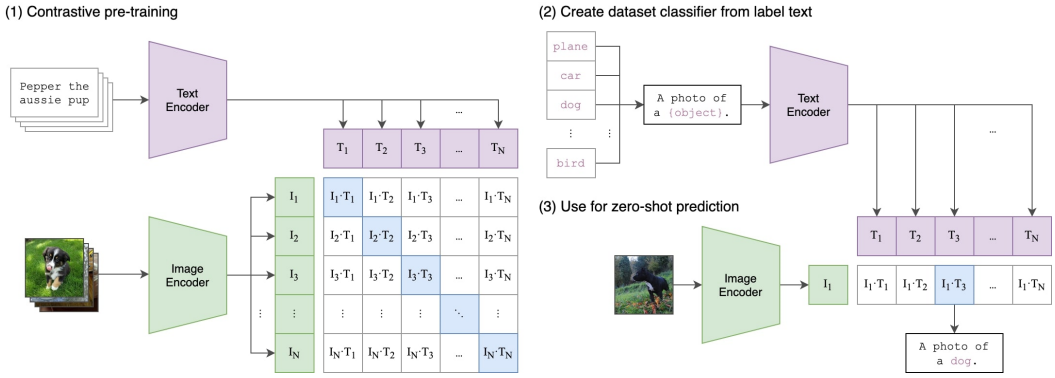


Figura 1.2: Struttura di allenamento ed utilizzo del modello CLIP[57].

Allenamento e applicazioni

Per il pre-training il modello CLIP[57] i ricercatori di OpenAI si sono quindi basati, secondo la tecnica del contrastive learning, su un semplice task di allenamento, corrispondente alla corretta associazione tra testi ed immagini, andando a sfruttare un dataset di ben 400 milioni di elementi scaricati dal web. In particolare, l'addestramento avviene considerando insiemi (batch) di N coppie, derivando per prima cosa gli embedding per ciascuna modalità, mappandoli quindi nello spazio di destinazione e calcolando infine la matrice di similarità tra tutte le possibili combinazioni testo-immagine come rappresentato nel primo passaggio della figura 1.2.

L'obiettivo del modello, allenato in modo end-to-end, diviene quindi quello di massimizzare il valore degli elementi presenti sulla diagonale di tale matrice, corrispondenti alla somiglianza tra gli coppie di testi ed immagini effettivamente corrispondenti, minimizzando invece i rimanenti. Per ottenere tale comportamento gli autori hanno optato per la funzione di costo contrastiva della cross-entropia simmetrica, definibile come

$$\mathcal{L} = -\frac{1}{2N}\left(\sum_{i=1}^{N}\log p_{\text{image2text}}(i|i) + \sum_{i=1}^{N}\log p_{\text{text2image}}(i|i)\right)$$

dove $p_{\text{text2image}}(i|j)$ rappresenta la probabilità che un testo i venga associato ad un immagine j, derivata attraverso l'applicazione di una funzione softmax, e similmente $p_{\text{image2text}}$ corrisponde alla probabilità che un immagine i venga associato ad un testo j.

Completato il pre-allenamento, tale modello incapsula quindi un buon livello di conoscenza, sfruttabile ad esempio per risolvere una problema di classificazione di immagini, anche in modo zero-shot, ossia senza un ulteriore finetuning. In particolare, come schematizzato nei passaggi 2 e 3 della figura 1.2, è possibile rappresentare le diverse classi direttamente attraverso delle semplici formulazioni in linguaggio naturale andando poi a selezionare quella con la maggior similarità rispetto all'input visivo.

Oltre allo specifico task di classificazione, gli autori hanno riscontrato come il modello risultasse adattabile ad una vasta gamma di altri problemi (downstream task), tra cui anche quello del retrieval inter-modale tra immagini e testo, grazie alla somiglianza con gli obiettivi dell'alle-

namento. In particolare, gli esperimenti eseguiti hanno mostrato come CLIP potesse raggiungere risultati allo stato dell'arte nel caso del retrieval zero-shot, mantenendosi competitivo rispetto ad alcuni modelli con finetuning specifici sui dataset considerati.

A riprova dell'efficacia di tale architettura e del vision-language pre-training, si può prendere in considerazione anche il framework BLIP, introdotto con l'articolo "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation"[39]. Tale framework, mirando a superare i limiti degli approcci precedenti sia dal punto di vista delle architetture, spesso difficilmente trasferibili tra task di generazione e task di retrieval, che dal quello dei dati, di qualità spesso sub-ottimale a causa del rumore introdotto dallo scraping web, estende infatti l'approccio seguito da CLIP per ottenere delle performance stato dell'arte su una serie di problemi riguardanti linguaggio naturale ed immagini.

In particolare, gli autori introducono due novità, basando il pre-allenamento su tre obiettivi distinti: image-text contrastive learning (analogo a CLIP), image-text matching trattato come un problema di classificazione binario, e generazione di un testo a partire da un'immagine (image-conditioned language modellig), e aggiungendo una seconda fase di addestramento basata su un dataset sintetico realizzato sfruttando il modello pre-allenato. Così facendo riuscirono a migliorare i risultati ottenuti su diversi dataset di benchmark, avanzando in particolare lo stato dell'arte nell'ambito del retrieval di immagini basato su query testuali.

1.1.2 Retrieval di video

Come mostrato anche negli articoli riguardanti CLIP[57] e BLIP[39], la capacità dei modelli basati su CLIP di analizzare congiuntamente testi ed immagini, può essere applicata anche a problemi riguardanti i video come nel caso della video action recognition o del text-to-video retrieval. Tali dati, infatti, sono costituiti da una sequenza di frame, ciascuno elaborabile attraverso le tecniche sviluppate per le immagini. La difficoltà aggiuntiva è però dovuta alla dimensione temporale, che deve essere considerata e gestita per poter ottenere dei risultati soddisfacenti.

Sulla base di queste considerazioni sono stati recentemente sviluppati diverse architetture capaci di ottenere ottimi risultati nell'ambito del retrieval dei video. Di seguito si presentano brevemente alcuni esempi di tali modelli.

CLIP4Clip

Una delle architetture più note in tal senso è CLIP4Clip ("CLIP For video Clip Retrieval") [42], frutto di una collaborazione tra alcuni ricercatori della Southwest Jiaotong University di Chengdu e di di Microsoft, e sviluppato come variante di CLIP per migliorare l'allora stato dell'arte nell'ambito del video-text retrieval. In particolare, gli autori scelgono di sfruttare la conoscenza già codificata nel modello CLIP grazie al suo ampio pre-allenamento, adattandola alla nuova tipologia di dato grazie ad un ulteriore addestramento su un ampio dataset di video, così da istruire il modello rispetto all'informazione temporale.

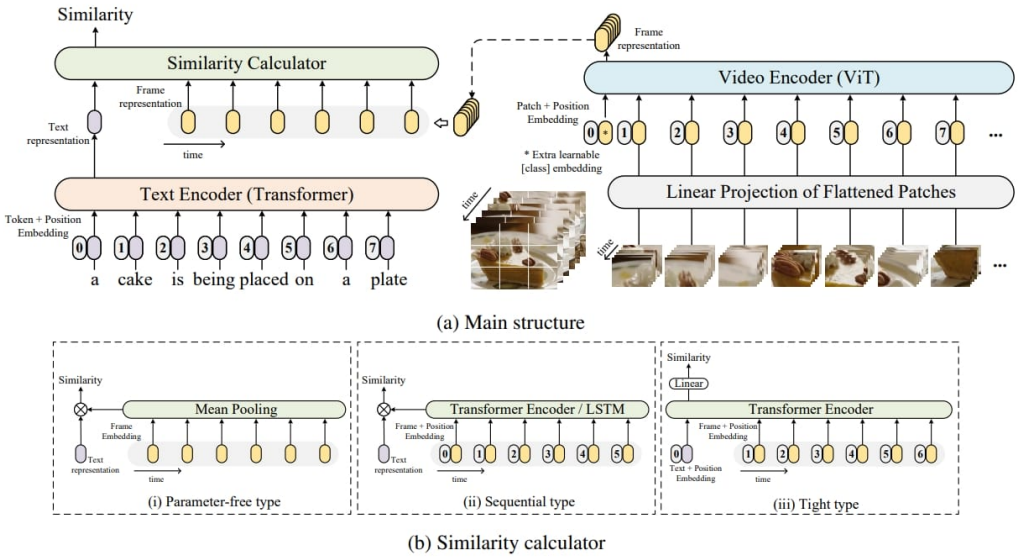


Figura 1.3: Architettura del modello CLIP4Clip con le relative varianti per l'unione delle feature dei frame e il calcolo della similarità con il vettore di feature testuali [42]

Per poter fare ciò gli autori hanno provveduto a modificare l'architettura originale in modo da poterla applicare a sequenze di immagini, come raffigurato nella porzione superiore della figura 1.3. Osservando tale struttura, si può notare come la codifica dei singoli frame sia analoga al caso di CLIP, mantenendo le medesime fasi di: suddivisione in patch 16×16, proiezione in vettori monodimensionali ed elaborazione attraverso i blocchi di un vision transformer. Allo stesso modo anche la fase di estrazione di una feature dall'input testuale ricalca la struttura di CLIP precedentemente descritta. Le differenze emergono invece nella fase di confronto delle feature appartenenti alle diverse modalità, col fine di poterne calcolare la similarità, elemento base dell'allenamento secondo la tecnica del contrastive learning.

In particolare gli autori hanno sperimentato con tre diverse strategie atte a unire in un unica feature l'informazione presente nei diversi frame, come riportato nella parte inferiore della figura 1.3. Una prima metodologia, detta "parameter-free", sfrutta un operazione di mean pooling sui frame per ottenere un vettore "medio" da confrontare con quello testuale mediante la similarità del coseno. Una seconda strategia ("sequential type") mira invece a includere anche l'informazione temporale, ignorata nel caso precedente, prima di unire le rappresentazioni dei frame come nel caso precedente. In particolare gli autori hanno optato per valutare delle reti neurali ricorrenti di tipo LSTM oppure, alternativamente, una rete basata sull'architettura del transformer. L'ultima alternativa, "tight type", prevede invece l'utilizzo diretto di un transformer sull'intera serie di frame concatenata con l'informazione testuale, in modo da permetterne un interazione più profonda, lasciando alla rete il compito della previsione della relativa misura di similarità.

Sulla base dei diversi esperimenti di video retrieval eseguiti dagli autori, la strategia generalmente migliore per i dataset di piccola, dimensione è risultata essere la prima, nonostante la maggior semplicità rispetto alle altre e la mancanza di un meccanismo per includere l'informazione sull'ordinamento temporale dei frame. Come spiegato nella sezione 2.2.1, tale configurazione

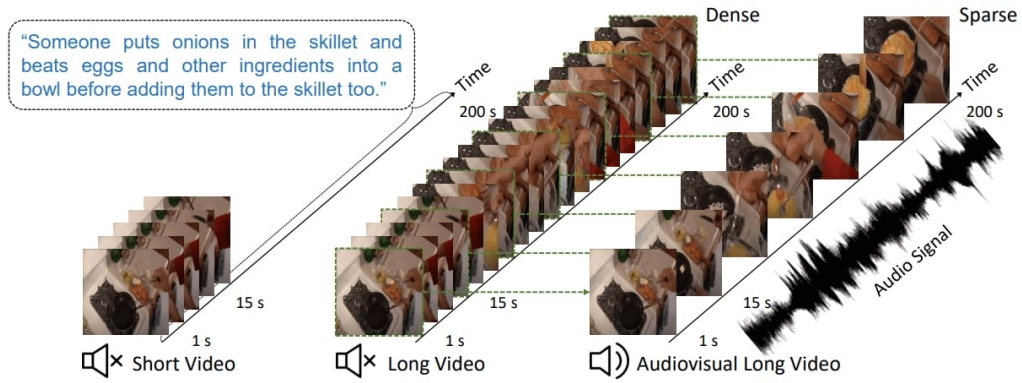


Figura 1.4: Confronto tra sampling dei frame denso, utilizzato da modelli come CLIP4clip, e sparso, utilizzato da ECLIPSE[41] insieme all'informazione audio, per video lunghi.

sarà anche una di quelle selezionate nell'ambito di questo lavoro.

ECLIPSE

Nonostante gli avanzamenti introdotti da CLIP4Clip[42], questo modello soffre di due limitazioni intrinseche. Per prima cosa, analogamente ad altri sistemi di video retrieval, è pensato per essere applicato a video di lunghezza piuttosto limitata, aspetto necessario per tenere sotto controllo la complessità computazionale relativa all'elaborazione della componente visiva. In questo caso infatti è possibile limitare il numero di frame da utilizzare come rappresentazione dell'intero video, pur mantenendo un buon contenuto informativo.

Un seconda problematica consiste nel fatto che i modelli come CLIP4Clip[42] ignorano le informazioni contenute nelle tracce audio degli elementi di input, rischiando quindi ottenere delle rappresentazioni solamente parziali dei dati.

Per superare tali aspetti alcuni ricercatori della University of North Carolina di Chapel Hill hanno proposto il modello ECLIPSE (Efficient Long-range Video Retrieval using Sight and Sound)[41]. L'idea di base di questo lavoro, rappresentata in figura 1.4, consiste nell'andare a utilizzare proprio l'informazione contenuta nella traccia audio come una "rappresentazione compressa" e complementare alla traccia video, permettendo quindi di ridurre il numero di frame necessari per raggiungere lo stesso livello di prestazioni.

In particolare gli autori hanno proposto di utilizzare un encoder audio con architettura ResNet-18[27] pre-allenato su VGGSound[13], un noto dataset audio-visuale utilizzato ad esempio per task di audio recognition, per ottenere delle codifiche vettoriali per ciascuno spezzone di audio da poter poi elaborare congiuntamente alle feature visive nel relativo codificatore. Per fare ciò hanno modificato i blocchi della self-attention del Vision Transformer in modo da includere due ulteriori meccanismi di attenzione, così da permettere alle informazioni visive di arricchire le feature dei video e vice versa. Una rappresentazione di tale meccanismo verrà riportata nella sezione 2.2.1, ed in particolare nella figura 2.8.

Un vantaggio di tale approccio è costituito dal fatto che l'inclusione di questi blocchi di attenzione audio-visuale aggiuntivi non impone di dover ripetere l'allenamento completo del modello. Infatti, utilizzando i pesi pre-allenati di CLIP4Clip[42] ed inizializzando a zero i layer di proiezione W^0 dell'output di tali sotto-blocchi, è possibile escludendone l'effetto, ricadendo di fatto nella struttura del modello originale. Come verificato dagli autori, è quindi possibile eseguire un ulteriore allenamento questa struttura su dati comprensivi dell'informazione uditiva in modo da aggiornare tali matrici di peso per poterle al nuovo scenario.

Nel paper originale vengono quindi riportati diversi risultati che evidenziano come ECLIPSE[41] possa sia in grado di superare le prestazioni di CLIP4Clip[42] su diversi benchmark di retrieval video, risultando contemporaneamente più efficiente sia in termini di memoria utilizzata che in termini di tempo di elaborazione, grazie al minor numero di frame necessari. Per tali ragioni, come spiegato nella sezione 2.2.1, si è deciso di considerare anche tale modello nell'ambito di questo lavoro, come alternativa a CLIP4Clip[42].

TEFAL

Di recente, un altro lavoro in ambito video retrieval che tiene in considerazione anche l'informazione sull'audio è stato proposto da un gruppo di ricercatori dell'Università di Amsterdam e di Amazon, con l'articolo "Audio-Enhanced Text-to-Video Retrieval using Text-Conditioned Feature Alignment"[31]. In particolare, partendo dai risultati presentati nel paper di ECLIPSE[41], gli autori notano come la strategia seguita in tale lavoro rischi di sfruttare le tracce acustiche in modo sub-ottimale rispetto al task del text-video retrieval.

Nell'articolo si evidenzia infatti come le informazioni contenute nelle componenti uditive e visive di un video possano essere complementari tra di loro, ad esempio qualora ci fossero di voci o suoni fuori campo. In questo caso però il meccanismo di attenzione audio-visuale di ECLIPSE[41], basandosi sulla similarità tra audio e video, rischierebbe di non prendere in considerazione tali elementi aggiuntivi a causa della loro scarsa correlazione, nonostante questi possano essere presenti e rilevanti per la query testuale.

Per superate questo ostacolo gli autori propongono quindi una variante dell'architettura di CLIP4Clip[42] con tre codificatori distinti per audio, video e testo, corrispondenti rispettivamente all'encoder di AST[25], un modello di basato su transformer utilizzato per compiti di classificazione audio, e agli encoder già utilizzati in CLIP[57]. Le feature audio e video sono quindi elaborate separatamente tra loro, ma congiuntamente a quelle delle descrizioni, attraverso dei moduli di attenzione audio e video testuali. Così facendo è possibile ottenere delle rappresentazioni finali di audio e video condizionate dall'informazione testuale, permettendo al modello di mantenere le informazioni anche riguardo ad elementi non strettamente correlati tra le componenti uditive e visuali delle clip.

Tali rappresentazioni vettoriali vengono quindi combinate attraverso un operazione di somma e confrontate con la feature della query testuale secondo la tecnica dell'apprendimento per con-

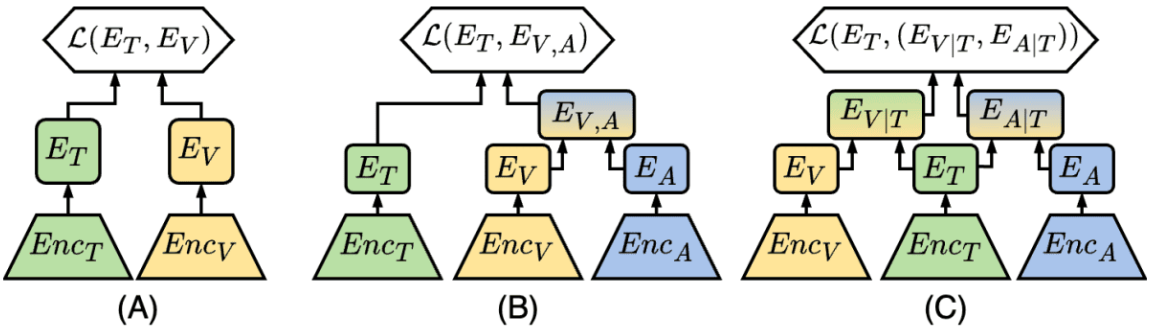


Figura 1.5: Confronto schematico tra l'architettura utilizzata in CLIP4Clip[42] (A), ECLIPSE[41] (B) e TEFAL[31] (C)

trasto. La differenza di tale strategia rispetto a quelle seguite in CLIP4Clip[42] e in ECLIPSE[41] è raffigurata schematicamente in figura 1.5.

Come mostrato dagli esperimenti degli autori, il modello TEFAL[31] riesce quindi a migliorare ulteriormente i risultati dello stato dell'arte per quanto riguarda il text-video retrieval, ponendosi come valida alternativa ai precedenti. Nonostante ciò, in questo lavoro si è scelto di non prenderlo in considerazione, per questioni sia legate alle tempistiche, che ai risultati già soddisfacenti ottenuti dagli altri modelli, lasciandolo piuttosto come possibile sviluppo futuro. Si nota comunque come l'approccio di TEFAL[31] potrebbe risultare particolarmente adatto alla natura dei video di natura artistica utilizzati in questo lavoro (si veda la sezione 2.1.1), a causa dell'editing svolto dagli artisti che potrebbero andare ad introdurre volontariamente una forma di dissonanza tra gli elementi visivi e la traccia audio, al fine di comunicare il proprio messaggio artistico.

1.1.3 Retrieval di ambienti 3D

Se i problemi di retrieval riguardanti immagini e video hanno riscosso molta attenzione da parte del mondo della ricerca, lo stesso non si può dire del caso del retrieval applicato a scenari tridimensionali. Tale area di ricerca, pur risultando promettente anche grazie alla crescita del Metaverso , risulta infatti ancora agli albori, specie per quanto riguarda il caso dell'associazione a descrizioni testuali.

I primi lavori in tale are infatti si sono concentrati sull'associazione di scene tridimensionali e immagini 2D [5, 6, 79], mentre più di recente sono stati pubblicati dei lavori che prevedono di allineare delle descrizioni testuali con una rappresentazione point-cloud[78], ossia costituita da una serie di punti nello spazio, basata su una rappresentazione a grafo dei relativi elementi costituenti [4, 2], o costituita da una collezione di immagini [3, 1] per tali ambienti complessi. Nell'ambito di questo lavoro ci si concentrerà su quest'ultimo caso, andato a seguire una strategia di rappresentazione simile come spiegato nella sezione 2.2.2.

Più in particolare, limitandosi ai lavori relativi alle ultime due strategie di codifica degli ambienti, si possono individuare due ambiti di applicazione. Il primo, indicato dagli autori col termine text-apartment retrieval, è legato al recupero, attraverso query testuali, di scenari 3D

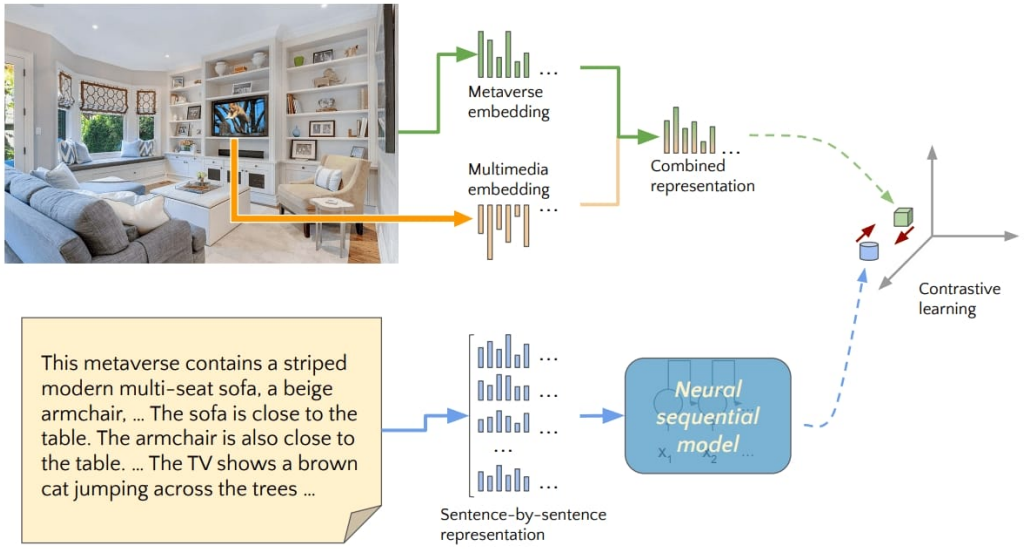


Figura 1.6: Architettura di alto livello proposta per il retrieval di scenari nel Metaverso contenenti dei video [2]

raffiguranti degli appartamenti[1, 3], con lo scopo ultimo di facilitare gli utenti finali nella ricerca di una accomodazione adeguata alle proprie esigenze, superando i limiti dei convenzionali sistemi di ricerca utilizzati in ambito immobiliare.

Un secondo campo di applicazione, coincidente con l'ambito di questo lavoro di tesi, si concentra invece sul retrieval di scenari tridimensionali appartenenti al Metaverso, contenenti degli elementi multimediali come video [2] o immagini di stampo artistico [4].

Retrieval di ambienti 3d contenenti video

Più nello specifico nel caso del primo di questi lavori proposti da Abdari et al., in cui viene anche definito il problema del retrieval text-to-Metaverse[2] come produzione di un ordinamento di Metaversi m_i rispetto ad una query testuale d_j, dato un dataset di coppie $\{D\} = \{(m_i, d_i), i \in \{1..N\}\}$, si prende in esame il caso di ambienti tridimensionali casalinghi contenenti dei video riprodotti attraverso dei televisori integrati nella scena. Gli autori propongono quindi di associare a ciascuno di tali ambienti una descrizione testuale creata sulla sulla base dei metadati assegnati originariamente alle scene e ai modelli 3D degli oggetti inclusi in esse, creando un dataset da utilizzare per il problema di retrieval analizzato.

L'architettura proposta, rappresentata in figura 1.6, segue quindi la falsariga degli approcci presentati in precedenza, componendosi di una fase di estrazione delle feature per i dati appartenenti alle diverse modalità, seguita da una loro rielaborazione e confronto secondo la tecnica del contrastive learning. In particolare, ciascuna scena, codificata come un grafo degli oggetti che contiene, e ciascun video vengono elaborati attraverso due Variational Auto Encoder pre-allenati per ottenerne le corrispondenti rappresentazioni vettoriali. Queste vengono quindi concatenate e ulteriormente processati attraverso una rete di tipo feed forward per ottenere il vettore corrispondente all'intera scena.

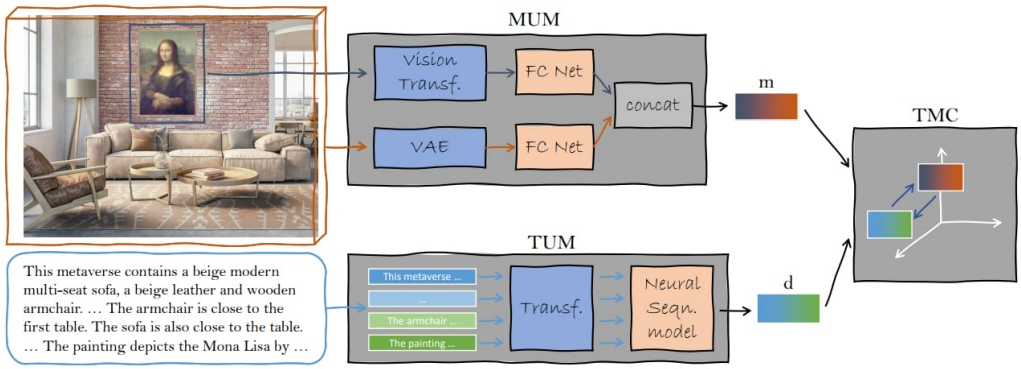


Figura 1.7: Architettura di alto livello proposta per il retrieval di scenari nel Metaverso contenenti dei quadri [4]

Per quanto riguarda le descrizioni testuali, invece, vista la notevole lunghezza dovuta alla loro verbosità, si è proposto di suddividerle in una sequenza di frasi più corte, vedendo ciascuna descrizione come una collezione di frasi. Ciascuna di queste viene quindi codifica separatamente attraverso un modello BERT per ottenere una sequenza di feature da poter unire in un embedding finale mediante una rete neurale ricorrente bidirezionale basata su unità GRU.

Analogamente ai casi precedenti, tali rappresentazioni possono quindi esser confrontate sulla base della loro similarità. In particolare, gli autori hanno utilizzato la funzione di costo triplet loss, comunemente scelta nell'ambito di task di retrieval.

Retrieval di ambienti 3d contenenti immagini

Un altro lavoro interessante e particolarmente correlato al lavoro di questa tesi è stato recentemente presentato con l'articolo "A Language-based solution to enable Metaverse Retrieval"[4]. In questo lavoro gli autori affrontano nuovamente il problema del text-to-Metaverse retrieval, utilizzando però degli scenari 3D relativi ad appartamenti contenti dei quadri di noti artisti come elementi multimediali.

In particolare, l'architettura proposta, rappresentata in figura 1.7, è simile a quella utilizzata nel caso precedente, prevedendo le medesime tecniche di estrazione delle feature sia per quanto riguarda le scene, che per le descrizioni testuali. Per estrarre, invece, le rappresentazioni delle opere d'arte contenute negli ambienti è stato utilizzato l'encoder visivo derivato dal modello CLIP[57], basato sull'architettura del Vision Transformer[17]. Gli autori propongono quindi di elaborare ulteriormente le feature delle scene e dei quadri attraverso delle reti di tipo feed-forward, concatenandone i risultati al fine di ottenere la rappresentazione finale da porre a confronto con quella testuale come in [2].

Un'ulteriore differenza rispetto al caso precedente è costituita dalla dimensione del dataset utilizzato. Se infatti in [2] gli autori aveva optato per un dataset relativamente ridotto di circa 3400 elementi, in questo caso è stato utilizzato un numero dati dieci volte superiore.

In ogni caso, pur costituendo dei passi importanti verso la trattazione del retrieval text-to-Metaverse, in entrambi gli articoli si sottolinea come la mancanza di altri dataset relativi a questo scenario costituisca un limite all'avanzare di questa nuova area di ricerca.

1.2 Dataset per il retrieval automatico

Come è noto la qualità dei risultati ottenibili dai modelli di deep learning è soggetta alla disponibilità di opportuni dati su cui allenare i modelli. Per tale ragione in questa sezione si presenteranno brevemente alcuni dei dataset comunemente utilizzati nell'ambito del retrieval automatico, con particolare attenzione rispetto all'ambito di questo lavoro di tesi. In particolare si manterrà la struttura utilizzata per la sezione precedente, trattando prima i dataset relativi ad immagini e video, porgendo particolare attenzione al caso di elementi legati al mondo dell'arte, e concludendo quindi con quelli utilizzati nell'ambito delle scene 3D.

Dataset per il retrieval di immagini

Il mondo dei dataset per il retrieval di immagini si può suddividere in due macro-categorie: quelli realizzati medianti un processo di annotazione manualmente e quelli ottenuti mediante tecniche semi-automatiche di web scraping. Le principali differenze possono essere associate a qualità e quantità dei dati, con i primi che risultano generalmente più curati, mentre quelli del secondo tipo puntano sulle maggiori dimensioni, sacrificando la precisione delle annotazioni.

Esempi di quest'ultima tipologia sono dataset quali Conceptual 12M (CC12M)[12], rilasciato nel 2021 da Google e contenente 12 milioni immagini corredate da delle descrizioni testuali estratte dalle pagine HTML di estrazione, LAION-400M [64], dataset open source composto da 400 milioni di coppie testo-immagine. Tali insiemi di dati vengono tipicamente utilizzati per eseguire un primo allenamento dei modelli di grandi dimensioni, come CLIP [57], in modo da fagli acquisire delle ottime capacità di base applicabili poi a diversi scenari.

Nel caso dell'addestramento di modelli di dimensioni inferiori o per includere dati di qualità migliore nel pre-allenamento, come avviene ad esempio nel caso di CLIP [57], si possono utilizzare i dataset di dimensioni ridotte ma maggiormente curati. Alcuni esempi comunemente utilizzati nell'ambito del retrieval di immagini sono: MS-COCO [40] comprendente più di 120000 immagini con diverse descrizioni testuali associate a ciascuna di esse, Visual Genome [36], contenente più di 100000 immagini corredate, oltre alle descrizioni, da una serie di informazioni aggiuntive relative agli oggetti raffigurati e alle loro relazioni, e Flickr30k [56], contenente all'incirca 30000 immagini principalmente riguardanti persone o animali e derivato dall'omonimo sito.

Per eseguire il finetuning dei modelli su compiti di retrieval più specifici si utilizzano invece dei dataset ridotti riguardanti un argomento ben preciso e tipicamente annotati in modo manuale o semi-manuale. Per lo scenario di applicazione di questo lavoro di tesi risultano particolarmente interessanti le raccolte di dati di matrice artistica, di cui esistono diversi esempi.

Tra questi possiamo distinguere due tipologie principali: quelli che associano a ciascuna opera una descrizione testuale del contenuto visivo e quelli che si focalizzano invece sulle emozioni suscitate nello spettatore. Nella prima categoria rientrano ad esempio SemArt[24], realizzato da dei ricercatori della Aston University e corredato da una serie di metadati riguardanti la tipologia e lo stille delle opere, e Artpedia[68], rilasciato dai ricercatori dell'università di Modena e Reggio Emilia.

Per quanto riguarda i dataset focalizzati sul rapporto emotivo tra spettatore ed opera si possono citare Artemis v1 [8] e v2 [50], contenenti 439000 annotazioni collezionate attraverso un processo di crowdsourcing e costituite dall'emozione principale provata dall'annotatore insieme ad breve commento a riguardo, e il dataset ArtELingo[49] che estende l'approccio del precedente ad uno scenario internazionale, aggiungendo annotazioni in arabo e cinese.

Dataset per il retrieval di Video

Passando allo scenario del text-to-video retrieval si può osservare nuovamente una suddivisione tra dataset di grandi dimensioni, più generalisti ed utilizzati per il pretraining dei modelli di grandi dimensioni e dataset di dimensioni minori sfruttati per scopi di finetuning o per valutare le prestazioni dei modelli in scenari più specifici. Per la prima tipologia alcuni noti esempi sono costituiti da HowTo100M[48], contenente oltre 100 milioni di video istruttivi ed utilizzato per il pre-training di CLIP4Clip[42], e YouTube-8M[7] che colleziona all'incirca 8 milioni di video dalla piattaforma di Google.

Per quando riguarda l'altra categoria di dataset, utilizzati ad esempio come benchmark per confrontare le prestazioni dei diversi modelli proposti in letteratura, possiamo trovare una vasta gamma di possibilità differenti. Alcuni di questi sono direttamente organizzati in clip annotate, come nel caso di Charades [66] che consiste in circa 10000 clip da 30 secondi raffiguranti persone che recitano lo svolgimento di attività quotidiane, mentre altri sono costituiti da video di lunghezza maggiore ma con descrizioni associate a specifici istanti temporali, in modo da permetterne l'utilizzo anche in compiti di localizzazione temporale delle scene di interesse. Esempi di questi ultimi sono ActivityNet [35], comprensivo di oltre 20000 video estratti da Youtube, e DiDeMo[28], contenente circa 10000 video postati su Flickr.

Tali dataset possono anche essere sfruttati per ottenere delle annotazioni relative agli interi video, semplicemente concatenando le diverse descrizioni associate ad una certa clip. Tale approccio viene seguito ad esempio in dagli autori di ECLIPSE[41] su diverse collezioni di dataset, come QVHighlights[37] contenente circa 3000 video caratterizzati da una durata media intorno agli 8 minuti, per poter valutare il retrieval anche in uno scenario caratterizzato da video di durata relativamente elevata.

Oltre alle distinzioni sopracitate, è anche possibile trovare delle collezioni di video fortemente legate ad un argomento specifico. Un esempio di ciò è costituito dal caso di YouCook2 [81], improntato sull'annotazione temporale di video di cucina ottenuti da YouTube.

Ciò nonostante, per quanto riguarda l'ambito di applicazione di questo lavoro di tesi, non risultano al momento disponibili dataset relativi a video di natura artistica, rendendo dunque necessario un lavoro di ricerca ed annotazione ex-novo, come verrà descritto nella sezione 2.1.

1.2.1 Dataset per il retrieval di ambienti 3D

Passando infine analisi del panorama di dataset riguardanti il retrieval di ambienti tridimensionali, non sono ancora presenti delle vere e proprie collezioni di riferimento che li associno a delle descrizioni testuali. Come si è già evidenziato nella sezione 1.1.3, infatti i lavori iniziali in ambito di retrieval di scene 3D prevedevano l'associazione tra scene 3d ed immagini 2D. In particolare, nell'ambito della competizione SHREC [5, 6, 79] si prevedeva l'utilizzo di dataset che associassero modelli tridimensionali a delle immagini fotografiche o raffiguranti dei disegni, sulla base di una serie predeterminata di categorie.

Per quanto riguarda invece i lavori riguardanti il compito di text-to-scene retrieval, nel caso di [78] gli autori propongono un dataset denominato CRISP (Cross-modal Retrieval on Indoor Scenes Point-cloud) composto da 10000 scene associate a delle corrispondenti descrizioni testuali e alle rappresentazioni point-cloud, mentre i lavori basati sul compito di text-apartment[1, 3] sfruttano un dataset pre-esistente di scenari 3D riguardanti stanze ammobiliate [23] associandovi delle descrizioni testuali derivate in base ai metadati degli oggetti contenuti in esse. Quest'ultimo approccio viene utilizzato anche nei lavori riguardanti il text-to-metaverse retrieval[2, 4], sfruttando il medesimo dataset di partenza e andando ad inserire nelle scene degli elementi multimediali estratti da altri dataset, come ad esempio YouCook2[81] nel caso dei video.

Per quanto riguarda l'ambito di applicazione di questo lavoro di tesi, non risultano al momento disponibili dataset relativi a scenari tridimensionali relativi a esibizioni d'arte o a musei virtuali. Di conseguenza, come indicato nella sezione 2.1.3 si è optato per una realizzazione da zero delle scene, in modo da poter avere il pieno controllo sul processo di generazione.

2

Metodologia e implementazione

In questo capitolo si esporranno i passaggi seguiti per affrontare il problema dell'associazione ordinata di esposizioni virtuali d'arte multimediale rispetto ad una richiesta testuale di un utente, trattando gli aspetti di metodologia e esponendo i principali dettagli implementativi. La trattazione sarà suddivisa in due sezioni principali riguardanti, rispettivamente, la definizione e generazione dei dataset, e la scelta dei modelli di intelligenza artificiale utilizzati, affiancata da una descrizione della loro architettura.

La prima sezione sarà ulteriormente suddivisa per trattare separatamente: gli aspetti di creazione di un dataset contenente video di natura artistica, la scelta del dataset per le immagini raffiguranti opere d'arte, la realizzazione degli spazi espositivi virtuali e la generazione del dataset finale riguardante le esposizioni complete.

La seconda sezione verrà anch'essa suddivisa in sottosezioni in modo da presentare separatamente i modelli di intelligenza artificiale utilizzati per eseguire il retrieval dei soli video artistici e successivamente quello delle esposizioni d'arte complete.

2.1 Definizione dei dataset

Il primo passo per poter affrontare il problema dell'associazione di esibizioni virtuali d'arte multimediale, contenenti in particolare immagini e video, consiste nella scelta o creazione di un dataset contenente una serie di scenari virtuali associati a delle relative descrizioni testuali. Come sottolineato nel capitolo precedente, ad oggi la diffusione di tale tipologia di dataset riguardante scenari tridimensionali complessi risulta ancora estremamente scarsa, e tale osservazione si applica in particolar modo nel contesto di applicazione di questo lavoro, per il quale non sono ancora presenti dataset di riferimento.

Parte di ciò è probabilmente dovuto al fatto che l'esposizione del pubblico a tale tipologia di scenari virtuali è ancora relativamente limitata, rendendo ridotta la richiesta. Nonostante ciò, grazie al crescente interesse verso il mondo del Metaverso[46], dovuta in parte anche alla recente esperienza della pandemia sembra suggerire uno sviluppo futuro anche nell'ambito dello scenario

analizzato in questo lavoro di tesi. A supporto di tale intuizione il ministero della cultura e del turismo cinese riporta che durante il capodanno lunare del 2020 i musei della regione avessero trasmesso in rete oltre 2000 diverse visite virtuali[26].

Per sopperire alla mancanza di un suddetto dataset e permettere quindi di studiare l'applicazione di tecniche di retrieval per il caso d'uso scelto, si è quindi partiti dalla raccolta dei dati necessari. In particolare si è deciso di cominciare collezionando una serie di video a stampo artistico da poter poi impiegare nelle fasi successive anche come base per la selezione dei dataset relativi alla rappresentazione tramite immagine delle opere d'arte.

2.1.1 Selezione dei video

Il primo aspetto da considerare per reperire i video necessari consiste nella definire di cosa si debba intendere per "video artistico". Infatti una tale formula potrebbe assumere diverse sfumature di significato a seconda dei casi. Un primo scenario potrebbe essere quello di una registrazione di una qualche performance artistica, come ad esempio un'opera lirica o teatrale. Il commento di un critico o di un artista che descrive una creazione potrebbe essere un'alternativa altrettanto valida, e lo stesso vale per un'opera cinematografia come un film o un cortometraggio.

Nel caso specifico di questo lavoro, con il termine "video artistico" o formulazioni analoghe si intenderà piuttosto un elemento appartenente ad una delle sue seguenti due categorie:

- Videoarte: in inglese "video art", forma d'arte nata negli anni '60, nella quale i video sono intesi come vere e proprie opere d'arte dove gli artisti adottano un "linguaggio artistico basato sulla creazione e riproduzione di immagini in movimento mediante strumentazioni video" [74]. Tale forma artistica si distingue dal cinema o da sue sotto-categorie, in quando non aderisce necessariamente alle classiche convenzioni che definiscono questi ultimi, come ad esempio la presenza di una trama ben definita o l'impiego di attori, risultando in una forma espressiva più libera [74];

- Arte performativa concettuale: in inglese "performance art", ossia registrazioni di artisti che eseguono una performance in presenza o meno di un pubblico. Come nel caso precedente, questa branca si distingue dalla categorizzazione più generale di arti performative a causa dell'assenza di aderenza stretta ad una serie di convenzioni generalmente adottate in queste ultime, come la presenza di un copione predefinito da recitare in molteplici occasioni, e il ruolo di primo piano assunto dai concetti alla base di una opera rispetto al suo risultato estetico. Esempi di tali lavori sono le performance di Marina Abramović o di Jackson Pollock [73].

Tale scelta nasce dalla volontà di cercare di rimanere quanto più aderenti possibile alle opere che si potrebbe trovare rappresentate in una vera e propria galleria d'arte reale, come ad esempio nel caso di una esposizione sulla videoarte o della Biennale d'arte di Venezia, e dalla facilità di reperimento dei dati. Sebbene infatti in una esposizione possa essere comune trovare anche video

a carattere più informativo, contenenti per esempio delle precisazioni dell'artista sulla propria visione artistica o su un'opera in particolare, e l'inclusione di tali elementi sia stata presa in considerazione nella fase preliminare, si è infine deciso di non considerarli.

Le motivazioni alla base di tale scelta sono diverse. Un primo aspetto è costituito dalla difficoltà di reperimento di elementi corrispondenti ai requisiti richiesti. Si è infatti notato come i video di tale tipologia non vengano spesso associati a metadati rilevanti e in particolare manchino di un corrispondente paragrafo testuale che ne descriva il contenuto, se non in modo estremamente generico. Tale aspetto ne limita quindi l'applicabilità al nostro caso d'uso a meno di non applicare delle tecniche di estrazione più sofisticate. Un esempio in tal senso potrebbe essere costituito dall'estrazione di una trascrizione testuale del parlato di tali video, unita ad un sistema automatico atto a riassumerne i contenuti principali.

Un ulteriore ostacolo è rappresentato dal fatto che i video a carattere divulgativo trattano simultaneamente svariate opere, anche ponendole a confronto le une con le altre. Tale aspetto, unito al precedente, rende particolarmente complessa la corretta associazione con i relativi argomenti, necessario per la creazione di esibizioni semanticamente coerenti. Un'eccezione in tal senso è costituita da una collezione di dati della National Gallery of Art di Washington, che risultano pubblicamente accessibili e permettono di mettere in relazione tra di loro i diversi materiali artistici mediante un ricco sistema di metadati[51].

Un ultimo aspetto da considerare è legato alla durata di tale tipologia di contenuti. I modelli di intelligenza artificiale infatti vengono generalmente applicati a video di lunghezza limitata, a causa della crescente richiesta di potenza computazionale all'aumentare della dimensione dell'input. Per tale ragione, si è scelto di fissare un tetto di 15 minuti per ciascuno degli elementi da reperire.

Fonti

Chiarite la tipologia di elementi da ricercare e le caratteristiche di interesse si è passati alla ricerca di possibili fonti per reperire i video di interesse. In particolare, dopo aver analizzato alcune decine di siti candidati, è stata eseguita una forte selezione al fine di rientrare nei requisiti stabiliti. Ad esempio si è deciso di ignorare eventuali agglomeratori generici di video, come nel caso di piattaforme generaliste di streaming quali YouTube, Vimeo, e Internet Archive, o di biblioteche digitali di opere artistiche come Europeana. Infatti, sebbene tali candidati possano includere alcuni elementi validi rispetto ai requisiti richiesti, la loro ricerca appare eccessivamente dispendiosa nello scenario di questo lavoro a causa della vasta mole di dati. Un altro ostacolo comunemente incontrato è stato quello dell'impossibilità di accesso alle opere indicate in un archivio digitale, a causa della loro distribuzione solamente sotto licenza acquistata o per assenza di una loro copia in formato digitale.

Al termine di tale processo sono state selezionate 11 diverse fonti, raggruppabili in due categorie e descritte brevemente di seguito:

- archivi digitali contenenti collezioni di video di diversi artisti:
  - DIVA station [16]: sito web parte dell'archivio video digitale ospitato dal Centro di Arti Contemporanee di Lubiana e nato nel 2005 con intenti di ricerca, documentazione, preservazione e archivio di videoarte;
  - Kadist [33]: sito web legato all'omonima organizzazione non profit d'arte contemporanea fondata a Parigi nel 2006;
  - Magmart 100x100=900 [43]: sito web legato ad un festival internazionale di video arte nato nel 2006, ed in particolare si è considerata la sottosezione legata al progetto internazionale "100x100=900 (100 videoartists to tell a century)" realizzato nel 2013 per celebrare i 50 anni dalla nascita della video arte;
  - Ubuweb [71]: archivio online nato 1996 dal poeta Kenneth Goldsmith per contribuire alla diffusione dell'arte d'avanguardia e sperimentale;
- siti personali di artisti che contengono le relative opere come parte del proprio portfolio artistico. In particolare sono stati selezionati i seguenti:
  - Emily Alden Foster [18], artista statunitense;
  - Colette Copeland [15], artista statunitense;
  - Silvia De Gennaro [67], artista italiana;
  - Francesca Fini [22], artista italiana;
  - Ora Kolmanovsky [52], artista Israeliana;
  - Shahar Marcus [65], artista Israeliano;
  - Evelin Stermitz [18], artista Austriaca.

Ciascuna delle fonti indicate contiene diversi "video artistici" corrispondenti alla definizione precedentemente indicata e correttamente associati a delle descrizioni testuali riguardanti il loro contenuto visivo. Inoltre, ciascun elemento risulta generalmente corredato da alcuni metadati come il titolo dell'opera, il nome dell'artista e l'anno di realizzazione, potenzialmente utili per organizzare il dataset o arricchire il contenuto informativo dei testi associati ai diversi elementi.

Scraping ed organizzazione dei metadati

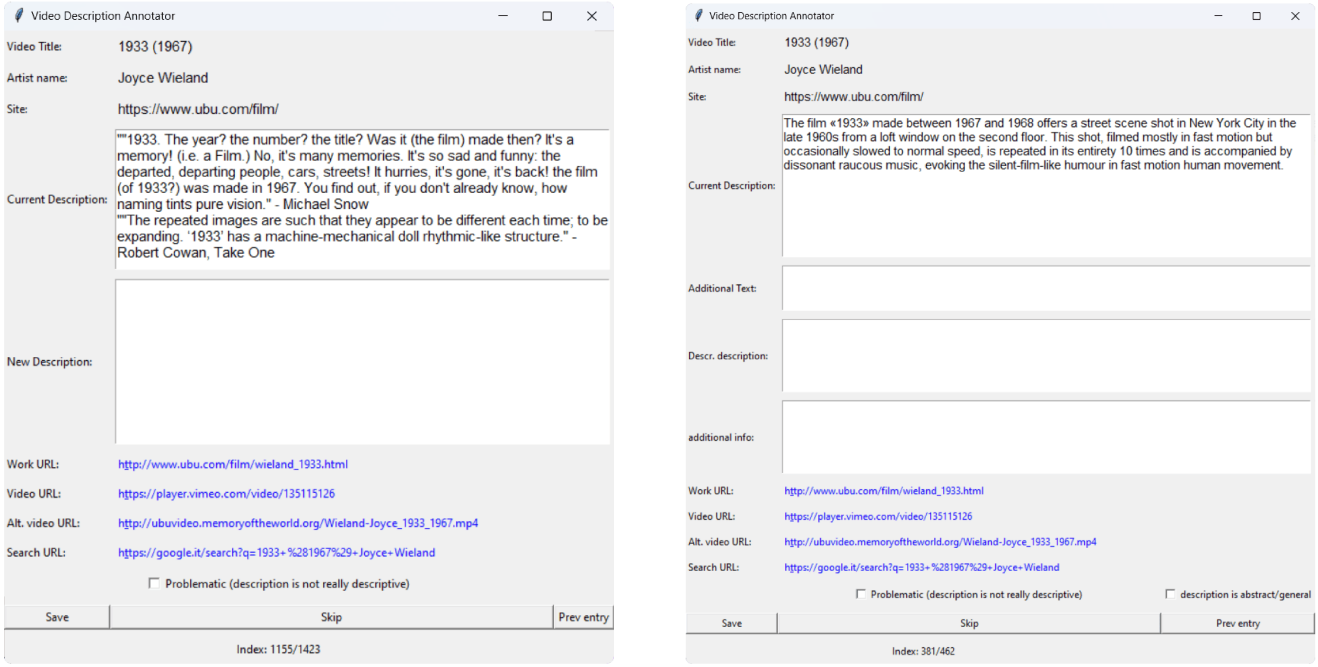
Al fine di recuperare i file video e i relativi metadati, si sono utilizzate delle tecniche di web scraping, adattandole di volta in volta alla fonte specifica. Ciascun sito presenta infatti una propria struttura ed organizzazione dei contenuti testuali e multimediali, richiedendo un forte intervento manuale.

Più nello specifico, dopo aver analizzato la struttura del sorgente HTML di per ognuna delle fonti, sono stati realizzati diversi script ad hoc utilizzando il linguaggio Python. Una libreria

particolarmente utile utilizzata in questo caso è Beautiful Soup [61] che permette di eseguire l'analisi sintattica (parsing) di pagine HTML al fine di estrarne le informazioni di interesse, poi organizzate in dei file testuali in formato JSON.

Al termine della raccolta dei metadati dalle diverse fonti si è proceduto alla loro unione in un unico dataset, contenente all'incirca 1600 elementi ed organizzato secondo una struttura comune. In particolare, a ciascun elemento sono stati associati i campi brevemente descritti di seguito, evidenziando quelli opzionalmente nulli:

- `title`: stringa testuale rappresentate il titolo dell'opera d'arte;
- `description`: stringa testuale rappresentate una testuale dell'opera d'arte;
- `work_url`: stringa testuale rappresentate url relativa alla pagina web utilizzata per l'estrazione dei metadati;
- `video_url`: stringa testuale rappresentate url al file video dell'opera d'arte;
- `video_id`: stringa testuale rappresentate un identificatore univoco per l'opera d'arte tra quelle considerate;
- `is_vimeo`: valore booleano per indicare se il `video_id` corrisponda ad un identificativo univoco per la piattaforma di streaming video Vimeo;
- `artist_name`: stringa testuale rappresentate il nome dell'artista;
- `duration`: valore numerico rappresentate la durata del video in secondi, estratto dal file video scaricato;
- `year` (opzionale): valore numerico intero rappresentante l'anno di realizzazione del video, quando indicato sul sito dell'opera;
- `extracted_year` (opzionale): valore numerico intero alternativo al precedente, utilizzato per rappresentare l'anno di realizzazione del video quando quest'ultimo non fosse esplicitamente citato nella pagina web con i metadati. In questo caso il valore è stato estratto dai metadati del file video e mantenuto separato in quanto rappresenta solamente una stima dell'attuale anno di realizzazione;
- `magmart_reference_year` (opzionale): valore numerico intero rappresentante l'anno di riferimento dell'opera video rispetto al progetto "100x100=900" per la fonte Magmart;
- `artist_nation` (opzionale): stringa testuale rappresentante la nazione di residenza dell'artista;
- `artist_bio` (opzionale): stringa testuale rappresentante una breve bibliografia dell'artista;



(a) Prima fase (b) Seconda fase

Figura 2.1: Interfacce grafiche utilizzate per la prima e per la seconda fase di pulizia dei metadati relativi ai video artistici

- `abbreviated_description` (opzionale): stringa testuale rappresentante una versione condensata del campo `description`;
- `tags` (opzionale): lista di stringhe testuali associate ad un'opera d'arte, esplicitamente indicate nella relativa pagina web o estratte dai metadati del video ospitati sulla piattaforma di streaming Vimeo;
- `work_type` (opzionale): stringa testuale rappresentante la categoria dell'opera d'arte, secondo l'eventuale classificazione presente nel sito di provenienza;
- `alternative_video_url` (opzionale): stringa testuale rappresentante un url alternativo al file video dell'opera d'arte;
- `additional_text` (opzionale): stringa testuale rappresentante un delle informazioni testuali aggiuntive sull'opera.

Completata l'organizzazione di tale dataset preliminare si è passati alla sua analisi per identificare gli elementi utilizzabili in base alla qualità delle descrizioni testuali associate loro.

Filtraggio e adattamento del dataset

Da un'analisi preliminare su un campione degli elementi è emerso infatti come in diversi casi le descrizioni associate ai video non fossero realmente descrittive rispetto al loro contenuto, contenendo piuttosto citazioni, commenti generici sulla vita o visione dell'artista o discussioni critiche. Un esempio di ciò è riportato in figura 2.1a.

Per tale ragione si è deciso di eseguire una prima scrematura degli elementi problematici dal dataset, andando ad eliminare nello specifico quelli per i quali i video non risultassero accessibili o fossero eccessivamente lunghi, e quelli dove la descrizione testuale non contenesse sufficienti aspetti descrittivi e non si riuscisse a reperire facilmente una descrizione alternativa. Per quest'ultimo caso, in particolare, di è deciso di adottare tale metodologia per evitare di allontanarsi eccessivamente dallo scenario di applicazione del lavoro proposto, in cui le descrizioni delle opere vengono curate da un esperto di dominio.

Trattandosi di un numero di elementi relativamente contenuto e vista la tipologia del compito, si è optato per un'analisi manuale dei singoli elementi, realizzando un semplice tool grafico per facilitare tale processo. Più nello specifico, si è utilizzato il linguaggio Python e la libreria grafica tkinter [53] per la realizzazione di una semplice interfaccia utente, visualizzabile nella figura 2.1a.

Il processo ha quindi previsto la lettura delle diverse descrizioni e il loro confronto, almeno in modo approssimativo, con i video pubblicamente accessibili, cercando di mantenere una politica di selezione abbastanza conservativa in modo da evitare un'eccessiva riduzione della dimensione del dataset e ridurre i già lunghi tempi di elaborazione. Una visione completa di tutte le opere visive avrebbe infatti richiesto oltre 145 ore, da sommare al tempo necessario per la lettura e confronto con le relative descrizioni.

Nonostante ciò, questa prima selezione ha richiesto una considerevole quantità di tempo, portando ad una riduzione di oltre il 70% degli elementi. Il dataset ottenuto ha mantenuto quindi all'incirca 460 opere, risultando però ancora piuttosto grezzo e richiedendo una seconda fase di elaborazione più approfondita.

Nonostante la presenza di una porzione descrittiva nei metadati selezionati, si è infatti notato come le descrizioni tendessero spesso a mescolare queste informazioni con nozioni di contorno, riguardanti la vita o visione artistica dell'autore. Similmente alla fase precedente si è quindi proceduto a passare nuovamente in rassegna gli elementi rimasti, utilizzando un semplice tool con interfaccia grafica simile al precedente, visualizzabile in figura 2.1b.

Durante tale processo, egualmente dispendioso in termini di risorse, si è andati ad isolare la porzione di descrizione effettivamente descrittiva dal resto del testo, cercando di integrarla con informazioni aggiuntive quando possibile. Non avendo a disposizione le competenze di un esperto di dominio, si è cercato di ridurre al minimo la modifica del contenuto originale, integrando le descrizioni con frasi estratte da commenti o critiche su ciascuna opera ed evitando per quando possibile la loro rielaborazione.

Nelle figure 2.1a e 2.1b è presente un esempio di tale trasformazione riguardante la video opera "1933" dell'artista canadese Joyce Wieland, visionabile sul sito Ubuweb[1]. In questo caso si può notare come si passi da una descrizione iniziale contenente delle recensioni dell'opera estremamente soggettive e focalizzate principalmente sui concetti che esprime, ad una descrizione estremamente più aderente al contenuto visivo.

[1] `https://www.ubu.com/film/wieland_1933.html`

Oltre a ciò si sono individuati e rimossi eventuali elementi problematici, sfuggiti alla prima fase di selezione, e si sono evidenziati quelli aventi delle porzioni descrittive ritenute eccessivamente generiche o astratte. Questi ultimi elementi, sebbene non selezionati per il dataset finale, potrebbero comunque essere presi in considerazione in lavori futuri per cercare di ottenere un dataset di dimensioni maggiori, interpellando un esperto di dominio per cercare di ottenere delle descrizioni adeguate.

Al termine di tale processo si è quindi ottenuto un dataset piuttosto ristretto di 218 elementi utilizzabili, e di 118 elementi da poter eventualmente tenere in considerazione per estensioni future. La struttura dei metadati rimane invariata rispetto a quella presentata precedentemente, sostituendo però al campo `description` la stringa di testo rivista, ossia relativa alla sola parte descrittiva rispetto al contenuto audio-visuale e ai significati ed emozioni espressi dall'artista. L'eventuale eccedenza del testo originale è stata invece spostata nel campo `additional_info`.

Normalizzazione del testo

Come ultimo passaggio, vista la diversità delle fonti utilizzate e la conseguente assenza di un formato univoco, si è eseguito un processo di normalizzazione delle stringhe testuali non relative a identificatori o pagine web. In particolare, si è optato nell'ordine per: un rimpiazzo dei caratteri di spaziatura ("whitespace charachters"), eventualmente ripetuti, con un singolo spazio semplice, seguito dalla normalizzazione dei caratteri Unicode in forma normale NFC e infine dalla rimozione di eventuali spazi in testa o in coda ("stripping").

Oltre a ciò, vista la presenza di segni diacritici dovuti alle diverse nazionalità degli artisti considerati, si è prodotta una variante dei metadati normalizzati in cui le lettere associate a tali segni sono state sostituite dalle corrispondenti "versioni alfabetiche base". Tali segni non sono infatti presenti nella lingua inglese, generalmente utilizzate per l'allenamento della maggior parte dei modelli di intelligenza artificiale pubblicamente accessibili.

Ricerca dei file video

Selezionati i candidati finali per il dataset, si è quindi passati al reperimento dei corrispondenti file video. In questo caso ci si è basati su download manager di terze parti, come jDownloader 2 [32] e yt-dlp [77], in modo da poter parallelizzare tali processi astraendosi dalla specifica piattaforma di distribuzione dei contenuti. Solamente nel caso della fonte UbuWeb, ci si è potuti basare in parte su del codice derivato dalla repository ubu24h [54], creata per trovare i link dei video della piattaforma e scaricarli automaticamente nell'ambito di una manifestazione culturale.

Definizione degli split

Conclusa la raccolta dei dati necessari si è passati alla definizione di una possibile suddivisione da utilizzare nelle successive fasi di allenamento. In particolare, vista la bassa numerosità del dataset si è optato per una suddivisione percentuale 75-5-25 rispettivamente per gli split di training,

evaluation e test, in modo da mantenere una buona quantità di elementi per allenamento e valutazione finale dei risultati.

Per mantenere bilanciate le diverse porzioni rispetto alla durata dei video, si è deciso di etichettare gli elementi in base a 3 intervalli di numerosità simile distinguendo tra contenuti corti (sotto i 3 minuti), medi (tra 3 e 6 minuti) e lunghi. Le proporzioni di opere per ciascuna categoria è stata quindi mantenuta durante la suddivisione andando a sfruttare i metodi di suddivisione "stratificata" offerti dalla libreria sklearn [55].

Oltre a ciò, avendo analizzato la distribuzione del numero di token associato a ciascuna descrizione testuale, utilizzando il tokenizzatore di CLIP, sfruttato dai modelli presentati nella sezione 2.2.1, e notando che all'incirca la metà degli elementi ricadevano al di sotto del relativo limite massimo di token, si è deciso di realizzare una partizione del dataset anche relativamente a tale sottoinsieme. Più in particolare, per evitare di produrre due suddivisioni completamente distinte e volendo mantenere il bilanciamento rispetto alla lunghezza si è optato per creare separatamente gli split per le due tipologie di elementi, unendole infine per creare gli split relativi all'intero dataset. Nel seguito questa sarà considerala la suddivisione principale, menzionando esplicitamente i casi in cui si dovesse utilizzare la sua versione ridotta.

2.1.2 Selezione delle immagini

Conclusa la selezione e il recupero dei video artistici si è passati alla selezione di un dataset adeguato per le immagini artistiche. Anche in questo caso si parte da una definizione del termine per chiarire gli elementi di interesse.

Con il termine "immagine artistica" o formulazioni analoghe si intenderanno delle immagini digitali raffiguranti delle opere quali quadri, dipinti, affreschi, pale d'altare o piccoli oggetti decorativi dipinti. Rimangono quindi esclusi da tale definizione elementi architettonici, sculture e installazioni, in quanto si è ritenuto che una loro rappresentazione tramite immagini non fosse adeguata per la loro corretta fruizione, in aggiunta a fotografia d'arte e arte digitale.

In questi ultimi due casi la scelta della loro esclusione è stata dovuta piuttosto alla mancanza di dataset su tali tematiche e alla quantità limitata di risorse dedicabili a questo progetto. Nonostante ciò la realizzazione e l'utilizzo di un tale dataset, potrebbe essere benefica per cercare di ottenere una maggiore omogeneità semantica tra video ed immagini selezionati per una esibizione, vista la maggior vicinanza temporale e la similarità di tecnologie impiegate con le opere video rispetto ai dipinti effettivamente impiegati nel seguito.

Definiti i soggetti di interesse sono state prese in considerazione diverse possibilità tra i dataset esistenti. In particolare, oltre al requisito minimo di possedere una stringa testuale descrittiva per ciascuna immagine, si era interessati in questo caso a disporre anche di metadati aggiuntivi, come autore, corrente artistica o tema, che permettessero di raggruppare semanticamente le diverse opere. L'idea di base infatti consiste nello sfruttare tali categorizzazioni per andare a selezionare

le immagini da includere in una esibizione virtuale rispettando un tema generale, come indicato meglio nella sezione 2.1.4.

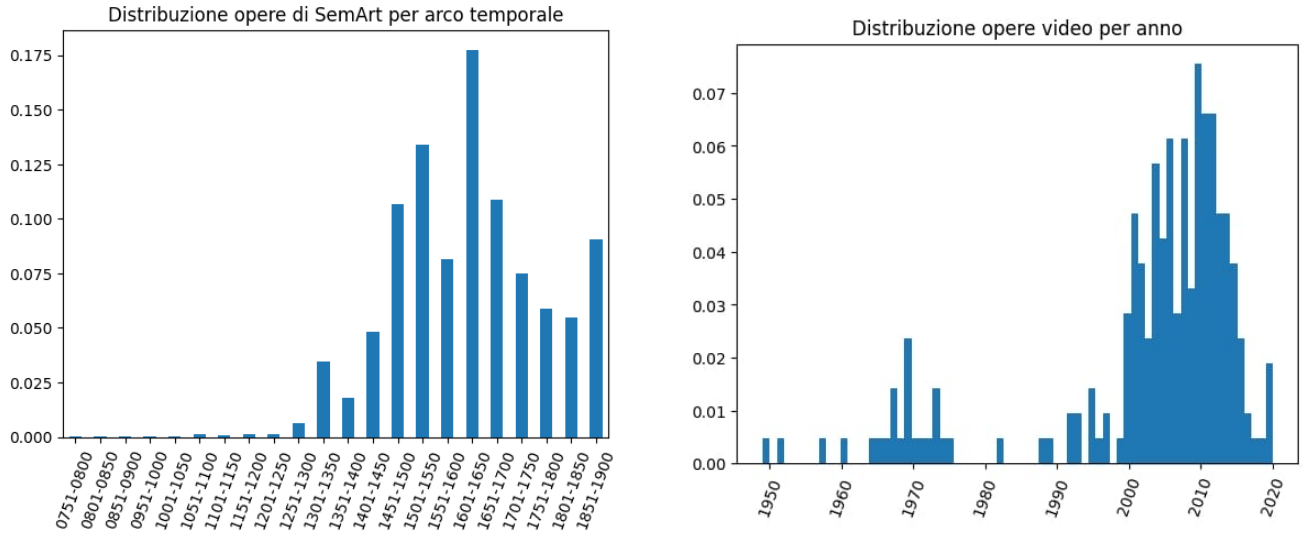
Come già menzionato nel capitolo precedente, ed in particolare nella sezione 1.2, fortunatamente esistono già diversi dataset riguardanti immagini di opere artistiche. Nel caso particolare di questo lavoro sono state prese in considerazione diverse alternative, di cui si riportano le principali caratteristiche nella tabella 2.1.

Tabella 2.1: Comparativa sui dataset di immagini artistiche presi in considerazione. Per la colonna "altri metadati" il numero tra parentesi indica il numero di diverse classi per il relativo raggruppamento.

dataset	numero opere	descrizioni testuali totali	descrizioni testuali per opera	tipologia descrizioni	altri metadati	periodo di rif. delle opere
SemArt [24]	21 384	21 384	1	descrizione visiva del contenuto, commenti su tecnica, autore o periodo storico	artista (3 281), scuola artistica (26), periodo storico (22), tipo (10), tecnica data	701-1900
Wikiart [62]	81 449	0	0	-	artista (1119), stile (27), genere (45)	1401-oggi
ArtEmis v1 [8]	80 031	454 684	almeno 5	spiegazione della scelta dell'emozione evocata all'annotatore	artista (1119), stile (27), genere (45), emozione dominante suscitata (9)	1401-oggi
Artpedia [68]	2 930	28 212	almeno 1	frasi che descrivono il contenuto visivo e frasi di contesto separate le une dalle altre	nessuno	1201-oggi

Dopo un un attenta analisi, la scelta finale è ricaduta sul dataset SemArt [24], realizzato da due ricercatori della "Aston University" di Birmingham e rilasciato nel 2018. Sebbene questo dataset non rappresenti rappresenti la scelta col maggior numero di opere, risulta piuttosto variegato e corredato da una buona gamma di metadati aggiuntivi, rendendolo particolarmente adatto al caso d'uso di questo lavoro.

Ciò nonostante, guardando alla colonna "periodo di riferimento delle opere" della tabella 2.1, e più in particolare andando ad analizzare la distribuzione delle opere rispetto all'anno di realizzazione, emerge una naturale problematica, condivisa almeno in parte da tutti i dataset analizzati: la discrepanza temporale tra video ed immagini artistiche. La maggior parte dei lavori infatti risulta prodotta in un periodo ben diverso rispetto a quello dei video artistici, la cui nascita si attesta negli anni '60, con la sola eccezione di opere contemporanee. Queste costituiscono però una porzione estremamente ridotta dei dataset analizzati impedendo quindi l'allineamento temporale delle due tipologie di opere. A supporto di tali osservazioni nelle figure 2.2a e e2.2b si mettono a confronto le diverse distribuzioni temporali delle opere contenute in SemArt[24] e nel dataset dei video.



(a) Prima fase

(b) Seconda fase

Figura 2.2: Confronto tra la distribuzione temporale delle opere video con quelle di SemArt.

Come si spiegherà in seguito nella sezione 2.1.4 questa problematica verrà affrontata sfruttando la similarità tra le descrizioni dei diversi elementi, al fine di cercare di ottenere delle esibizioni quanto più coerenti possibile dal punto di vista semantico.

2.1.3 Creazione degli ambienti virtuali

Conclusa la selezione dei dataset relativi a video e immagini artistiche si passa ora alla definizione della metodologia applicata per creare gli spazi d'allestimento virtuali in cui collocarle. Più nello specifico, come accennato nella sezione 1.2.1, vista la mancanza di dataset già pronti riguardanti degli spazi espositivi, e mirando a creare delle rappresentazioni realistiche, seppur semplificate, di ambienti in cui un un utente possa effettivamente immergersi per ammirare delle opere digitali, si è optato per una loro realizzazione da zero.

Per realizzare tali ambienti tridimensionali, ci si è quindi affidati al motore grafico Unity, comunemente utilizzato nell'ambito dello sviluppo di contenuti interattivi, come ad esempio video giochi o applicazioni di realtà virtuale, e programmabile attraverso il linguaggio C#.

Per affrontare in modo più sistematico la generazione delle scene 3D, si è optato per suddividere il problema in due fasi distinte. In particolare nella sezione 2.1.3 si tratterà della generazione degli spazi espositivi "spogli", ossia nei quali non sono ancora state inserite le opere d'arte virtuali. Successivamente, nella sezione 2.1.3, si discuterà dell'allestimento delle esibizioni vere e proprie, andando a collocare video e immagini all'interno degli spazi tridimensionali.

Generazione delle scene

Come accennato precedentemente, l'obiettivo di questa prima fase consiste nel generare degli spazi espositivi vuoti nei quali poter inserire in seguito delle opere artistiche. Per fare ciò,

si è partiti da del codice pre-esistente, sviluppato internamente al Laboratorio di Intelligenza Artificiale dell'università di Udine[2], in modo da accelerare le fasi iniziali di sviluppo. Su tale base si è quindi andati ad eseguire un ampio refactoring al fine di ottenere una struttura maggiormente estensibile ed aggiungere le funzionalità necessarie per lo scenario applicativo di questo lavoro.

Volendo chiarire la terminologia utilizzata nel contesto di questa tesi, si sottolinea come con il temine "spazio espositivo", talvolta sostituito alternativamente con quelli di "esposizione" o "museo", si indichi uno spazio tridimensionale organizzato secondo una serie di stanze, ciascuna delle quali può presentare dei muri piani da dedicare all'esposizione delle opere. Per semplicità, si è assunto che tali stanze fossero organizzate secondo una chiara successione lineare, agendo di fatto come un tunnel per un eventuale visitatore. Tale assunzione permette dunque di ottenere un naturale ordinamento delle stanze corrispondente al loro ordine di visita, nonché di associarvi una rappresentazione testuale che mantenga tale struttura per i paragrafi relativi alle diverse stanze.

Come si può notare in figura 2.3, si è deciso di mantenere le stanze degli spazi espositivi semplici e poco appariscenti. Questa scelta è motivata dalla volontà di focalizzare l'attenzione di un possibile visitatore, o di un modello di Intelligenza Artificiale, sulle opere esposte piuttosto che sull'ambiente circostante che potrebbe, altrimenti, costituire una fonte di disturbo.

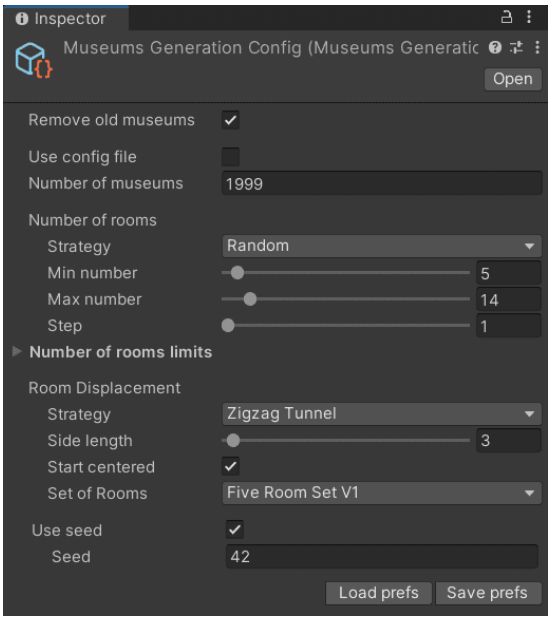
Per realizzare i singoli ambienti, si è fatto riferimento a delle risorse già disponibili, sfruttando in particolare il pacchetto "Apartment Kit" [69], pubblicamente disponibile nello store di Unity, che offre una serie di modelli tridimensionali di base, come muri, piastrelle e moduli per soffitti. Tali elementi possono essere combinati per creare una vasta gamma di ambienti, partendo da dei semplici spazi interni, sino ad arrivare a interi edifici o quartieri virtuali.

Nel contesto di questa tesi, si è optato per realizzare delle stanze a pianta quadrata aventi un unico ingresso ed un unica uscita su due lati distinti, ad eccezione della prima ed ultima, entrambe caratterizzate da una sola apertura. Nonostante ciò, il codice è strutturato in modo tale da risultare facilmente estendibile anche a stanze con conformazioni differenti tramite l'implementazione di un'opportuna interfaccia. In Unity, infatti, è possibile associare delle porzioni di codice (script) direttamente ai modelli tridimensionali, permettendo l'implementazione di funzioni appositamente studiate per essi.

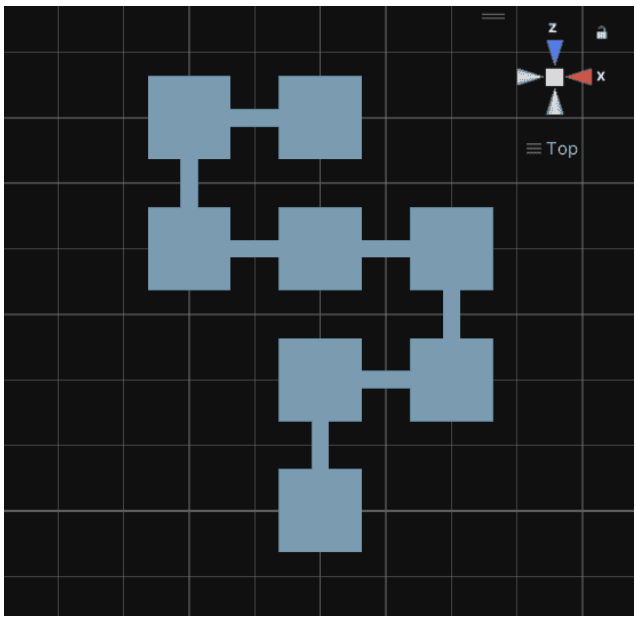
Per quanto riguarda lo spazio espositivo le suo complesso, come già accennato, le diverse stanze risultano logicamente organizzate secondo una struttura a catena. Nonostante ciò, si è cercato di rendere la loro disposizione spaziale più interessante e variata implementando delle regole per organizzarle secondo un percorso a serpentina, come quello rappresentato in figura 2.4b. Tale struttura è inoltre facilmente modificabile mediante una serie di impostazioni, accessibili mediante un piccolo pannello di controllo integrato direttamente nell'interfaccia grafica di Unity. In particolare, come si può notare dalla figura 2.4a , si è lasciato all'utente la possibilità di agire su numero di ambienti da generare, numero di stanze per ciascuno (nell'esempio raffigurato scelto

[2] `https://ailab.uniud.it/`

Figura 2.3: Interno del modello 3D di una stanza per le esposizioni



(a) Pannello impostazioni di generazione



(b) Vista dall'alto delle stanze

Figura 2.4: Schermate di Unity riguardanti il pannello di controllo per le impostazioni di generazione degli ambienti virtuali e la vista dall'altro della disposizione delle stanze per un museo di 8 stanze secondo una serpentina di lato 3.

in modo casuale rispetto ad alcuni parametri) e strategia di disposizione delle stanze nello spazio.

Nonostante questa interfaccia grafica permetta di controllare in modo interattivo le principali impostazioni di generazione, non permette una regolazione più puntuale delle caratteristiche del singolo museo. Per aggiungere una tale possibilità, utile nel caso della generazione automatica di un ampio numero di scene, si è deciso di aggiungere la possibilità di fornire le indicazioni sulle caratteristiche di ciascun museo attraverso un file di configurazione in formato JSON. Come si spiegherà nella sezione 2.1.4, tale file verrà conterrà tutte le informazioni riguardanti la generazione e decorazione dei singoli musei, permettendo così di disaccoppiare la fase di selezione delle opere che andranno a comporre una esibizione, dalla successiva realizzazione dell'esposizione virtuale vera e propria.

La fase di generazione produce quindi come risultato una serie di file con estensione "unity"

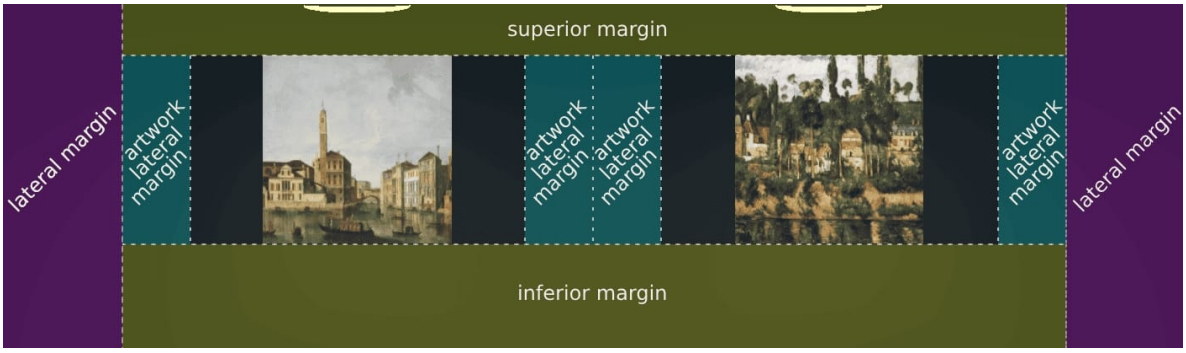


Figura 2.5: Schema di suddivisione degli spazi di una parete espositiva.

relativi alle diverse scene create, ciascuno associato a un nuovo file testuale in formato JSON. Lo scopo di tale file è quello di fungere da rappresentazione testuale della singola esibizione, contenendone le informazioni riguardanti la struttura e la serie di opere d'arte che dovrà ospitare, risultando utile come metodo di passaggio delle informazione alla successiva fase di decorazione.

Per poter valutare la qualità delle scene ottenute, a ciascuna scena si è aggiunto un semplice sistema di navigazione in terza persona, realizzato utilizzando gli strumenti messi a disposizione dal pacchetto "Starter Assets" e controllabile da tastiera. Attraverso tale sistema diventa quindi possibile visitare gli spazi tridimensionali realizzati consentendo, una volta terminata la fase di decorazione descritta nella seguente sezione, a simulare l'esperienza di visita di un possibile visitatore.

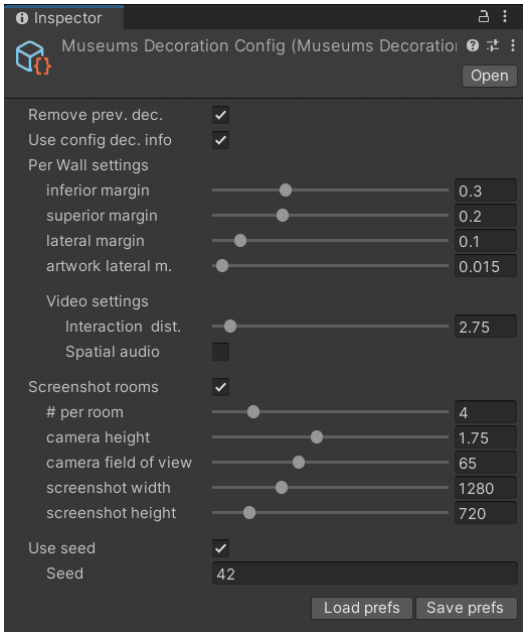
Decorazione delle scene

Dopo aver definito l'approccio per la generazione degli ambienti virtuali e realizzato il corrispondente codice, si è affrontato il problema della collocazione delle opere d'arte all'interno delle diverse stanze, indicato nel seguito col termine "decorazione". In particolare, ricevendo dalla fase precedente una scena tridimensionale relativa ad uno spazio espositivo e la lista di opere multimediali da inserirvi, si è interessati ad andare ad "esporle" sulle pareti espositive disponibili.

Per ciascuna di queste, si è deciso di organizzare lo spazio mantenendo dei margini vuoti lungo il loro perimetro e definire un ulteriore margine laterale tra due opere adiacenti. In questo modo è possibile ricavare l'area disponibile per ciascun elemento, entro la quale poterlo collocare nel rispetto delle proporzioni originali. Per chiarezza, la figura 2.5 riporta un diagramma raffigurante tale suddivisione degli spazi.

Analogamente alla generazione delle stanze, si è andati a realizzare un pannello di controllo interattivo anche per controllare tali impostazioni di decorazione. Una visualizzazione di quest'ultimo è presente in figura 2.6a, nella quale si possono vedere i valori utilizzati per i diversi margini. Oltre a questi sono visibili anche delle ulteriori opzioni per eseguire la cattura delle immagini degli interni, il cui ruolo verrà chiarito nella sezione 2.2.2.

Per quanto riguarda l'inserimento effettivo delle opere nelle scene tridimensionali si sono utilizzati dei semplici oggetti corrispondenti a dei piani su cui andare a visualizzare le opere come



(a) Pannello impostazioni di decorazione

(b) Vista interna di una stanza decorata

Figura 2.6: Rappresentazioni del pannello di controllo per la decorazione degli ambienti espositivi e la generazione dei relativi screenshot, e vista dell'interno di una stanza decorata con tre quadri a tema paesaggistico e il video "Still Life I" dell'artista Ana Sluga

delle texture nel caso delle immagini e attraverso l'utilizzo dei componenti "video player" messi a disposizione da Unity, nel caso delle clip. Per questi ultimi si è inoltre andati a implementare un semplice meccanismo di interazione basato sulla prossimità dell'utente.

In particolare, infatti, al fine di evitare un utilizzo eccessivo di risorse, ciascun video viene caricato solamente nel momento in cui un visitatore si avvicina e sosta di fronte allo schermo virtuale. Similmente la riproduzione viene automaticamente sospesa, e dopo un certo tempo riavvolta, qualora l'utente decidesse di allontanarsi dall'opera.

Prima di una di queste forme di interazione, ciascuno schermo virtuale risulta invece completamente bianco, mostrando le informazioni riguardanti autore e nome dell'opera. Tale scelta è volta a cercare di creare una chiara distinzione visiva tra i due diversi tipi di opere, sia per un eventuale visitatore virtuale, che per un modello di intelligenza artificiale (si veda Sezione 2.2.2). Un esempio di questa distinzione può essere osservato nella figura 2.6b, raffigurante una stanza decorata con tre quadri a tema paesaggistico e un video artistico.

Al termine della fase di decorazione si ottengono dunque le versioni aggiornate delle scene iniziali, complete delle opere d'arte disposte sulle pareti dedicate alla loro esposizione. Oltre a ciò, anche le rappresentazioni testuali in formato JSON associate a ciascun museo, vengono riviste in modo da includere l'informazione sulla stanza di esposizione di ciascun elemento artistico, al fine di facilitare la successiva fase di creazione del dataset finale.

2.1.4 Creazione del dataset delle esposizioni virtuali

All'interno del dataset costruito in questa fase sono presenti tre tipi principali di informazioni: le scene generate e decorate seguendo la procedura presentata nella sezione 2.1.3, le descrizioni testuali associate ad ogni museo, ed infine i quadri e i video collocati nelle stanze diverse. Se gli elementi testuali e le scene sono fondamentali per permettere di sviluppare modelli di Intelligenza Artificiale capaci di associare informazioni testuali a quelle visive, andando ad implementare applicazioni come ad esempio la generazione automatica di descrizioni testuali di ambienti complessi, o il recupero di esposizioni data una descrizione testuale, caso di studio analizzato in questa tesi, le informazioni sugli elementi multimediali si potrebbero considerare come ancillari.

Più in particolare, come spiegato nella sezione 2.2.2, tali elementi risulteranno utili per consentire ai modelli realizzati di porre maggiore attenzione al contenuto artistico degli elementi presenti nelle varie stanze. È tuttavia necessario sottolineare come tale scelta richieda delle assunzioni piuttosto forti rispetto in relazione all'interazione con gli ambienti virtuali.

Operare direttamente sugli oggetti artistici presenti in un'esposizione d'arte ospitata nel Metaverso, potrebbe infatti essere piuttosto problematico, visto che richiederebbe l'utilizzo di un sistema per il rilevamento degli oggetti (object detection) per individuare la posizione e tipologia delle diverse opere presenti. Benché infatti esistano sistemi adatti allo scopo, quali YOLO [59] e Faster RCNN [60], eventuali errori di rilevamento o di localizzazione delle opere potrebbe ridurre le prestazioni del modello di retrieval. Per tale ragione, l'assunzione di base di questo lavoro di tesi può essere vista come l'esistenza di un tale sistema ideale, ossia non affetto da errori.

Su queste basi è quindi possibile discutere l'approccio utilizzato per la creazione del dataset finale, la cui generazione dipende, come già accennato in più occasioni da un file di configurazione in formato JSON fornito in input nella fase di generazione.

Creazione del file di configurazione del dataset

In particolare, per semplicità, si è scelto di utilizzare uno script in Python per andare a definire tale configurazione partendo dalla decisione del numero totale di esibizioni, pari a 1999 elementi, e le percentuali di suddivisione per gli split di training, validation e test set, corrispondenti rispettivamente a 70%, 10% e 20%. Nel seguito parlando di un esibizione si intenderanno quindi i corrispondenti aspetti di configurazione inseriti nel file JSON.

Come precedentemente accennato, l'obiettivo della creazione del dataset consiste nel creare, seppur in maniera estremamente semplificata rispetto a quanto potrebbe fare un curatore d'arte, delle esposizioni tematiche sfruttando le categorizzazioni presenti nel dataset SemArt [24]. Per tale ragione, per ciascuna delle tre porzioni, la numerosità delle immagini è stata analizzata rispetto alle classi facenti riferimento a scuola artistica, periodo storico, tipo ed artista. Da tale analisi, i cui risultati sono riportati nella tabella 2.2, è emerso come il numero di opere per artista risultasse estremamente ridotto nel caso degli split di test e validazione. Per tale ragione, si è quindi deciso di escludere tale raggruppamento, focalizzandosi piuttosto sui rimanenti. In

	gruppo	split	numero medio di opere per categoria	numero massimo di opere per categoria	numero categorie con almeno 20 opere	numero categorie con almeno 30 opere
0	SCHOOL	train	679.81	7035	22	18
1	SCHOOL	val	48.09	422	7	6
2	SCHOOL	test	46.13	416	7	6
3	TYPE	train	1767.50	6683	10	10
4	TYPE	val	105.80	421	9	9
5	TYPE	test	106.10	401	9	7
6	AUTHOR	train	5.66	291	152	91
7	AUTHOR	val	1.75	21	1	0
8	AUTHOR	test	1.66	15	0	0
9	TIMEFRAME	train	841.67	3246	15	13
10	TIMEFRAME	val	66.12	205	12	11
11	TIMEFRAME	test	58.94	187	11	11

Tabella 2.2: Analisi della distribuzione delle immagini di SemArt[24] rispetto alle categorie presenti nei gruppi raggruppamenti riferiti a scuola artistica, periodo storico, tipo ed artista

particolare, avendo deciso di utilizzare un numero di stanze variabile tra 6 e 9, e ipotizzando di inserire due opere d'arte per ciascuna delle pareti, si è derivato il massimo numero minimo di immagini diverse per costituire un esposizione, pari a 34 elementi. Tale valore è stato quindi utilizzato come soglia per filtrare le classi da mantenere come scelta per le possibili tematiche, escludendo inoltre la categoria "others" relativa ai dati non categorizzati.

Si è quindi provveduto ad attribuire una tematica di riferimento a ciascuna esibizione del dataset finale. Tale assegnamento si è basato su una ripartizione equa tra le i diversi raggruppamenti, ossia scuola artistica, periodo storico e tipo, e proporzionale rispetto alla numerosità di ciascuna delle classi presenti in essi.

Definita una tale suddivisione, a ciascuna esibizione è stato assegnato un numero di immagini variabile derivato dal campionamento rispetto ad una distribuzione binomiale con probabilità di inserimento di un'immagine pari all'80% del numero di opere, e derivando un numero di video complementare. La scelta delle singole immagini è quindi avvenuta seguendo un campionamento uniforme senza reinserimenti rispetto al sottoinsieme di elementi appartenenti alla tematica selezionata e allo split di interesse, così da evitare sovrapposizioni di opere nelle tre suddivisioni del dataset finale.

Terminata tale fase si è passati all'associazione dei video a ciascuna esibizione, evitando tuttavia di suddividere il dataset in porzioni distinte di train, test ed evaluation a causa del numero ridotto di elementi a disposizione. In questo caso, visto che i video non risultano organizzati secondo la stessa gli stessi gruppi dei quadri, e non essendoci comunque, come già indicato in precedenza, una sovrapposizione temporale o stilistica tra le opere, si è deciso di assumere la similarità tra le descrizioni testuali come parametro alternativo di vicinanza tematica.

In particolare, si è deciso di codificare ciascuna descrizione utilizzando l'encoder testuale sfruttato dal modello di intelligenza artificiale CLIP [57], al fine di associare a ciascuna opera, immagine o video, un embedding che ne codificasse l'informazione semantica. Così facendo, diventa possibile associare un vettore numerico alla tematica di una specifica esposizione, ottenuto calcolando la media degli embedding delle corrispondenti immagini artistiche. Per scegliere quindi i video più rilevanti per ciascuna, è possibile confrontare tale rappresentazione con quella dei video artistici disponibili, utilizzando nel nostro caso la similarità del coseno, e selezionando gli elementi maggiormente affini.

L'applicazione naïve di tale idea, pur massimizzando la vicinanza tematica media tra video selezionati ed esibizioni, tendeva a sfruttare solamente una minima parte del dataset (circa il 15%). Per risolvere tale situazione sub-ottimale si è optato per una strategia alternativa, basata sempre sul concetto di similarità ma che permette di sfruttare l'intero catalogo di opere video. In particolare, per ogni esibizione sono stati calcolati i valori di similarità tra gli embedding di essa e dei diversi elementi video, convertendoli poi in probabilità attraverso l'impiego di una funzione softmax con parametro di temperatura pari a 0.12. Tali valori sono stati quindi utilizzati come parametri di probabilità di campionamento per una scelta casuale delle opere da inserire. Come valutato empiricamente, tale strategia, rispetto a quella naïve, ha il vantaggio di andare a scegliere effettivamente tutti i video presenti nel dataset, pur riducendo leggermente la similarità media dei video con i musei, che passa da 82.3% a 74.4%. Un esempio dei contenuti multimediali associati ad una stanza delle esibizioni generate è riportato nell'appendice A nella sezione A.1.

Questo passaggio conclude la generazione del file di configurazione relativo alle diverse esibizioni del dataset, permettendole l'utilizzo attraverso le procedure di Unity di generazione e decorazione per la creazione delle scene tridimensionali.

Generazione delle descrizioni testuali

L'ultimo passo della procedura per la creazione del dataset consiste nella generazione delle descrizioni testuali da associare alle esibizioni. Per fare ciò, ogni scena è stato associato un blocco testuale in lingua inglese basato sulla struttura degli ambienti e sulla opere esposte in ciascuna stanza. In particolare ci si è basati sul seguente modello, adattato di volta in volta al numero appropriato di stanze ed elementi artistici.

> This art exhibition has {`string with the cardinal number of rooms`} rooms other than the initial lobby.
>
> The {`string with the ordinal number of the room`} room contains {`string with the cardinal number of artworks`} artworks, {{{`string with the cardinal number of videos`} of which {"are videos" `if there is more than one video` "is a video" `otherwise`}} `if there is at least a video` "without any video" `otherwise`}.

> One of the room artworks is a {"video" `if it its a video artwork` "image" `otherwise`} artwork and has the following description. {`artwork description`}

Analogamente a quanto indicato nella, sezione precedente, un esempio più puntuale di descrizione testuale associato ad una stanza delle esibizioni generate è riportato nell'appendice A nella sezione A.1.

Per ricapitolare dunque la struttura finale del dataset, ciascun elemento è costituito da una esibizione d'arte virtuale corrispondente ad una scena di Unity, corredata dalla lista di immagini e video di opere d'arte esposte in ciascuna e associata ad una corrispondente descrizione testuale.

2.2 Selezione dei modelli di IA e delle strategie di elaborazione

Conclusa la trattazione sui dataset utilizzati, in questa sezione saranno esposti i modelli di Intelligenza Artificiale impiegati e sarà descritto l'approccio scelto per l'elaborazione dei dati. Seguendo la suddivisione della sezione 2.1, si partirà dai modelli applicati al caso del retrieval dei soli video artistici, volto a testare le prestazioni dei modelli sul dataset di video a nostra disposizione e a fornire informazioni utili in vista della successiva fase di applicazione al caso delle esposizioni complete.

2.2.1 Modelli per il retrieval dei video artistici

Per eseguire il retrieval dei video artistici, ci si è basati su due architetture distinte, entrambe basate sull'idea alla base del modello CLIP e adattandole al caso dei video. Infatti, il modello CLIP è stato rilasciato da OpenAI nel 2021 per affrontare il problema dell'allineamento tra immagini e testi associati. Nel seguito si esporranno separatamente i due modelli, evidenziandone le differenze e gli adattamenti effettuati per consentirne l'applicazione al dataset delle opere video.

CLIP4Clip

Il primo modello analizzato, CLIP4Clip ("CLIP For video Clip retrieval) [42], è stato sviluppato da ricercatori delle Southwest Jiaotong University di Chengdu, in collaborazione con ricercatori della Microsoft, e parte dall'idea del modello CLIP [57] per estenderla al caso dei video. In particolare, l'architettura si compone di due elementi distinti: un encoder visivo per elaborare i frame che costituiscono il video stesso, e un encoder testuale per trasformare la corrispondente stringa testuale in un embedding. Le rappresentazioni vettoriali dei diversi frame vengono quindi unite attraverso un'operazione di media componente per componente (mean pooling) e confrontati con il vettore del testo per calcolarne la similarità utilizzando il loro prodotto vettoriale interno, come mostrato nella figura 2.7.

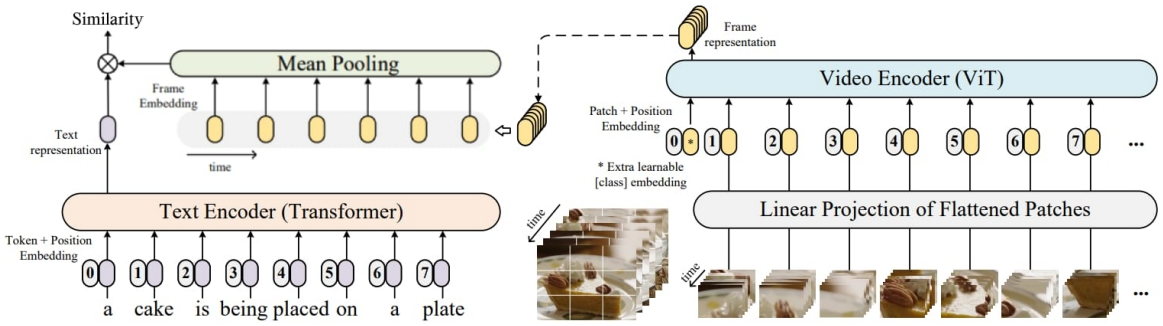


Figura 2.7: Diagramma rappresentante l'architettura del modello CLIP4Clip [42] con aggregazione delle feature visive attraverso mean pooling.

Questa struttura di elaborazione si può dunque applicare per l'allenamento del modello CLIP4Clip[42] analogamente a quanto avviene per CLIP[57], lavorando su batch di coppie (*video*, *descrizione t*
per calcolare una metrica di similarità su tutte le combinazioni possibili. Come nel caso della triplet loss, l'obiettivo consiste nel massimizzare la similarità degli abbinamenti corretti, corrispondenti ai valori presenti sulla diagonale principale, minimizzando invece i rimanenti. In particolare, la funzione obiettivo ottimizzata dagli autori mediante la tecnica della discesa del gradiente è la seguente:

$$\mathcal{L}_{v2t,t2v} = -\frac{1}{B}\sum_{i=1}^{B}\left(\frac{\exp\left(s\left(v_i,t_i\right)\right)}{\sum_{j=1}^{B}\exp\left(s\left(v_i,t_j\right)\right)}+\frac{\exp\left(s\left(v_i,t_i\right)\right)}{\sum_{j=1}^{B}\exp\left(s\left(v_j,t_i\right)\right)}\right)$$

dove B rappresenta la dimensione del batch, s_k e v_k gli embedding di testo e video del k-esimo elemento del batch, e s la funzione di similarità del coseno. In questo modo, si punta a risolvere simultaneamente sia il retrieval di un video a partire da una richiesta in formato testuale, sia lo scenario a parti inverse.

Nel contesto di CLIP4Clip[42], partendo dalla configurazione di pesi pubblicamente disponibili per CLIP[57], e seguendo quindi la tecnica del transfer learning, il modello è stato addestrato su dataset di grandi dimensioni contenenti milioni di video, in modo da introdurre conoscenza legata alla dimensione temporale. In questo lavoro, ci si baserà su questi ultimi pesi, nella versione con patch visuali si dimensione 32×32px, rilasciati pubblicamente dagli autori. In particolare, a partire da tale configurazione si andrà ad eseguire un finetuning del modello sul dataset di opere video in modo da cercare di ottimizzarne le prestazioni per questo specifico scenario di retrieval.

Analogamente alla procedura seguita dagli autori di CLIP4Clip[42] e al fine di rendere più efficiente la successiva fase di allenamento, si è provveduto a comprimere gli elementi di tale dataset ri-scalando la dimensione minore a 224px e riducendo il framerate a 3fps. Quindi, basandosi sul codice originale degli autori, è stato sviluppato il codice necessario per il caricamento del dataset in fase di allenamento e valutazione. Dopo alcuni test preliminari, si è deciso di estrarre i frame in modo uniforme rispetto ad un campionamento a 1fps, seguendo l'ordinamento originale dei video e considerando per ciascuno la porzione centrale di 224×224px analogamente a quanto suggerito in [42].

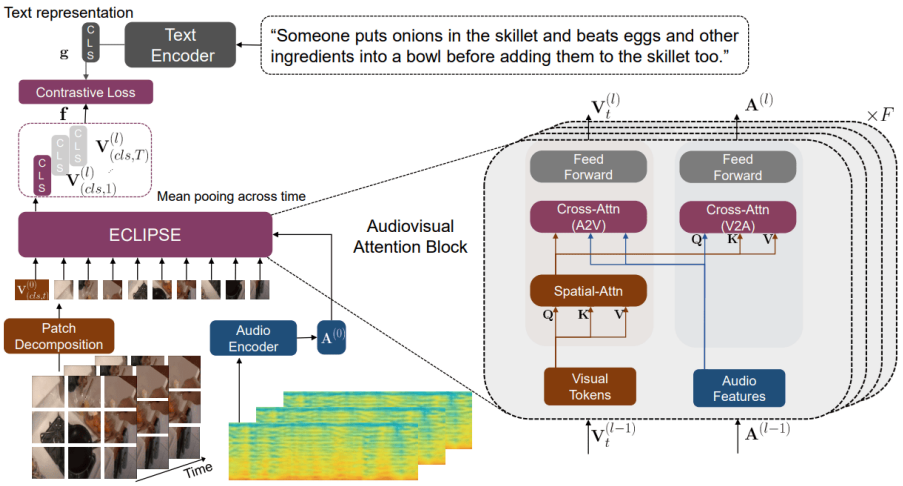


Figura 2.8: Diagramma rappresentante l'architettura del modello ECLIPSE [41] Nella rappresentazione dei blocchi interni all'encoder audio-video sono stati omesse per semplicità le connessioni residuali relative ai meccanismi di attenzione e alle reti "Feed Forward".

ECLIPSE

Nonostante le ottime performance dimostrate riportate dagli autori di CLIP4Clip[42] rispetto a vari benchmark per il retrieval di video, tale modello presenta una limitazione intrinseca: ignora completamente l'informazione uditiva associata a ciascun video. Per tale ragione, considerando che la traccia audio costituisce parte integrante dell'opera, contribuendo alla trasmissione del messaggio dell'artista, e avendo notato come tale informazione fosse spesso presente nelle descrizioni testuali associate ai video, si è scelto di considerare anche un'architettura che potesse gestire tale tipologia di dato.

In particolare, la scelta è ricaduta sul modello ECLIPSE ("Efficient CLIP with Sound Encoding") [41], proposto da alcuni ricercatori della University of North Carolina di Chapel Hill nel 2022. Tale architettura si basa sulla struttura di CLIP4Clip, aggiungendo però un encoder pre-allenato per elaborare il segnale audio e poterlo integrare nel meccanismo di attenzione dell'encoder visivo in modo da includere le informazioni sull'audio, come rappresentato in figura 2.8.

Osservando il diagramma, si può infatti notare come, tra il meccanismo dell'auto-attenzione "Spatial-Attn" ed il componente "Feed Forward", elementi classici dell'architettura Transformer [72], sia inserito un nuovo blocco chiamato "Cross-Attn (A2V)". Tale elemento applica la classica operazione di operazione di attenzione, definita come

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_h}}\right)V$$

utilizzando però come query le feature visive e come key e value quelle uditive, così da influenzare l'output del blocco attraverso l'informazione sull'audio. Si può inoltre notare che un blocco parallelo "Cross-Attn (V2A)" ricopre un ruolo complementare, permettendo alle feature visive di andare ad arricchire le l'informazione sull'audio.

Come accennato nella sezione 1.1.2, tale struttura permette di sfruttare la conoscenza già incapsulata dal modello CLIP4Clip[42] pre-allenato, richiedendo solamente un finetuning sul dataset di interesse per poter sfruttare l'informazione audio. Di conseguenza, anche in questo caso ci si è basati sui pesi di quest'ultimo modello per inizializzare il modello ECLIPSE[41], andando quindi ad eseguire un finetuning specifico sul dataset di nostro interesse.

Anche in questo caso, al fine di rendere più efficiente e rapido il processo di allenamento, si è andati ad eseguire una pre-elaborazione dei dati, gestendo separatamente gli elementi visivi da quelli uditivi. Per quanto riguarda il video si è nuovamente utilizzata una riduzione del framerate, portandolo a 3fps, mentre per l'audio è stata estratta la corrispondente traccia in formato wav.

Oltre a ciò, coerentemente con l'approccio seguito in [41], si è provveduto ad estrarre le corrispondenti feature sfruttando un encoder audio pre-allenato da VGGSound [13]. In particolare, per ciascuna traccia, si sono elaborati 64 spezzoni da 10 secondi ciascuno uniformemente distribuiti rispetto alla durata del video. A tal proposito, si fa notare come tale valore rappresenti solamente un limite superiore al numero di elementi da fornire in input al modello, mentre come indicato di seguito si utilizzeranno delle dimensioni inferiori.

Come in precedenza si è quindi provveduto ad implementare il codice necessario al corretto caricamento degli elementi precedentemente elaborati. In questo caso però, dopo alcuni test preliminari, si è optato per l'utilizzo di un numero di frame, e corrispondenti frammenti di audio, inferiore rispetto al caso di CLIP4Clip[42], testando valori pari a 8 e 16. Tale aspetto è strettamente legato all'idea di fondo seguita dagli autori di ECLIPSE[41] di utilizzare gli spezzoni audio come alternative "compresse" ad un campionamento più frequente dei frame già menzionata nella sezione 1.1.2.

Per i dettagli sull'allenamento si rimanda alla sezione 3.1.1 relativa agli esperimenti eseguiti e alla sezione B.2.2 dell'appendice B.

2.2.2 Modelli per il retrieval delle esposizioni d'arte multimediali

Conclusa la trattazione dei modelli utilizzati per il retrieval dei video artistici, si passa ora all'esposizione delle strategie proposte nel caso delle esibizioni virtuali. In particolare la figura 2.9 rappresenta una panoramica dello schema più generale delle architetture utilizzate nell'ambito di questo lavoro.

Tale struttura è suddivisibile in tre componenti principali: uno modulo per l'estrazione delle feature dalle esibizioni virtuali, un modulo per l'estrazione delle feature dalle descrizioni testuali e infine un meccanismo dedicato alla corretta associazione di tali rappresentazioni in uno spazio multidimensionale sulla base della definizione di un opportuno obiettivo che consenta di allenare un tale modello per il problema trattato. Come si può notare, tale approccio è analogo a quello adottato da molti metodi proposti per la risoluzione di problemi intermodali, come avveniva anche per i modelli trattai nella sezione precedente.

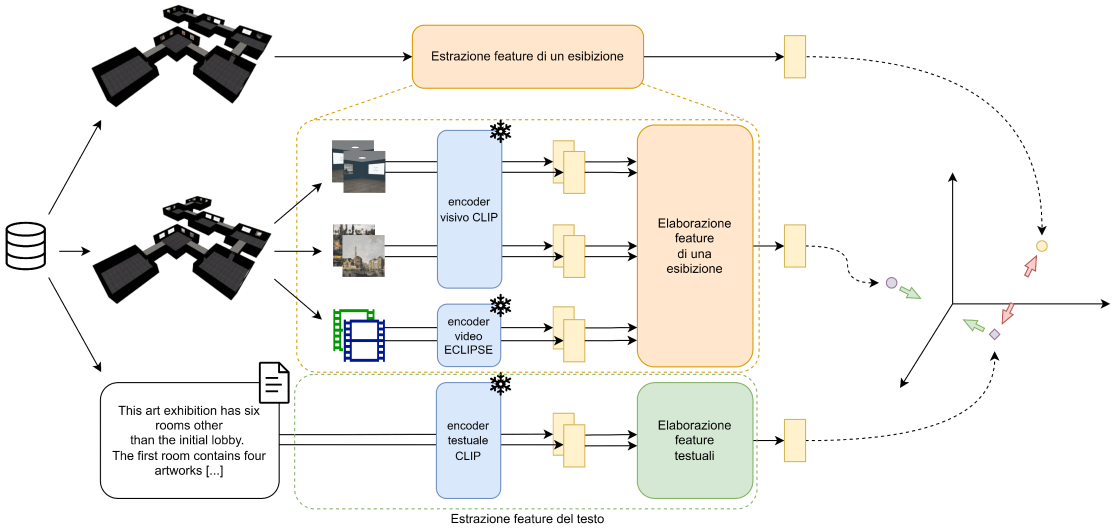


Figura 2.9: Panoramica dell'intera architettura utilizzata per elaborare i dati relativi al problema del retrieval di esposizioni data in input una descrizione testuale.

A differenza dei casi precedenti però, si può osservare come, il modulo per l'elaborazione delle esposizioni, sia suddiviso internamente in tre flussi di elaborazione. Il primo di questi costituisce il formato di rappresentazione scelto per le singole scene. Analogamente ai lavori di Abdari et al. [2, 4], si è scelto di rappresentare ciascuna stanza tramite quattro screenshot catturati in ognuna di esse utilizzando una telecamera rotante posta al centro. Oltre a tale informazione di base, come accennato nella sezione 2.1.4 si è deciso di considerare per ciascuna esposizione anche l'insieme degli elementi multimediali contenuti in essa, che vanno ad aggiungere due ulteriori flussi di informazioni per ciascuna scena.

Come sarà spiegato più nel dettaglio nelle sezioni 2.2.2 e 2.11b, i primi due flussi di informazioni, costituiti da immagini, saranno inizialmente elaborati attraverso CLIP[57], mentre per i video si utilizzeranno alternativamente i modelli CLIP4Clip[42] ed ECLIPSE[41], unendone successivamente le rappresentazioni attraverso un ulteriore blocco di elaborazione. Anche per quanto riguarda il testo si è optato per una elaborazione che sfrutta il modello CLIP[57], rimandando alla sezione 2.2.2 per ulteriori dettagli a riguardo.

Passando al meccanismo utilizzato per implementare l'obiettivo dell'allenamento, così da allineare le diverse feature nello spazio in modo che quelle degli elementi simili risultino vicine tra di loro e sufficientemente separate da quelle appartenenti a elementi dissimili, si è scelto di utilizzare funzione di costo la triplet loss [63]. Maggiori dettagli a riguardo saranno forniti nella sezione 2.2.2.

Una differenza sostanziale dell'approccio seguito in questo caso rispetto a quando eseguito nella sezione 2.2.1 per il finetuning di CLIP4Clip[42] ed ECLIPSE[41] consiste nella quantità di pesi aggiornati durante l'addestramento. Se infatti per questi ultimi modelli si è andati ad eseguire un allenamento di tipo end-to-end nell'affrontare il problema del retrieval dei video, che aggiorna cioè la totalità dei pesi disponibili, in questo caso si è optato per un più leggero approccio basato su feature (o feature-based). In particolare, questa strategia prevede l'utilizzo

di estrattori pre-allenati, i cui pesi sono mantenuti fissati, per ottenere delle rappresentazioni vettoriale iniziali dei dati di input. Durante il processo di allenamento si procede dunque ad elaborare degli ulteriori modelli di dimensioni ridotte, rappresentati nella figura 2.9 dai blocchi di elaborazione delle feature delle esibizioni e del testo, per ottenere l'allineamento delle feature desiderato.

Nelle seguenti sezioni, si vedranno più nel dettaglio le diverse porzioni dell'architettura descritta, partendo dalla strategia di elaborazione del testo, per poi passare a quella delle esibizioni d'arte. Per queste ultime, si esporranno due approcci distinti: uno "base", che non va a sfruttare l'informazione sull'organizzazione in stanze degli spazi espositivi, e uno "gerarchico", che analizza isolatamente le diverse stanze per derivarne delle rappresentazioni locali, da combinare successivamente in una rappresentazione globale del museo.

Elaborazione delle descrizioni testuali

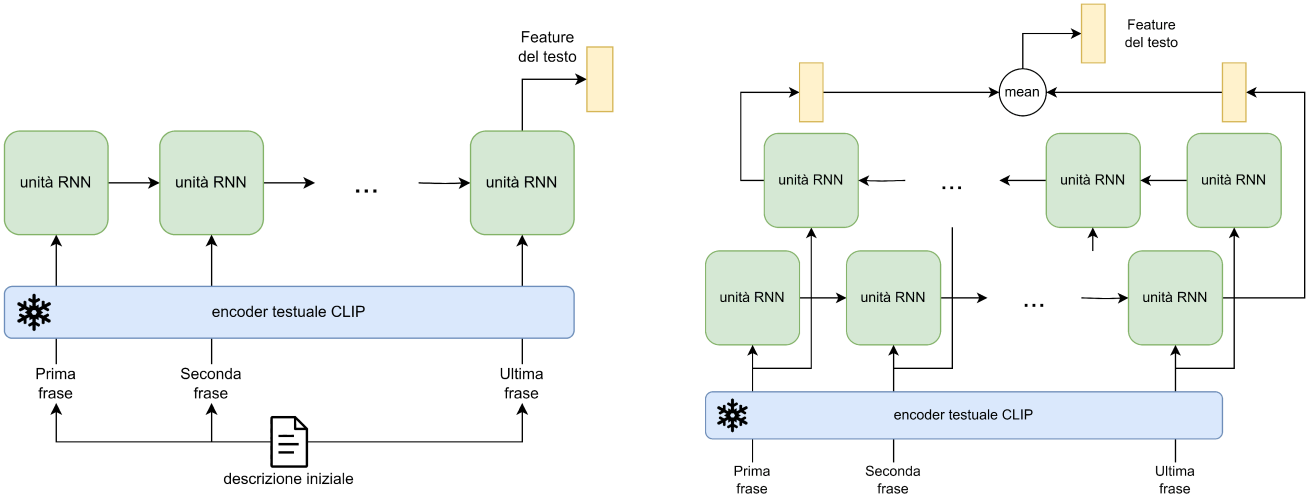
Considerando che le esposizioni sono formate da svariate opere, si può intuire come le corrispondenti rappresentazioni testuali possano crescere molto di dimensioni arrivando a comporre testi d formati da varie centinaia o migliaia di parole. Tali lunghezze risultano però molto superiori rispetto limite massimo al massimo supportato dall'encoder testuale di CLIP[57], impedendone una elaborazione diretta. Per questo motivo, seguendo nuovamente l'approccio di Abdari et al. [2, 4] si è deciso di adottare la seguente strategia di pre-elaborazione.

In particolare, il primo passo consiste nel suddividere ogni descrizione a livello dei singoli periodi, ottenendo per ciascuna esibizione una serie di stringhe testuali di dimensione contenuta. Ciascuna di queste, è stata poi processata mediante l'encoder testuale di CLIP [57] in modo da ottenerne una rappresentazione vettoriale. Il risultato di tale pre-elaborazione consiste dunque, per ciascuna delle esibizioni, in una serie di lunghezza variabile di vettori. Vista la struttura sequenziale, tale struttura suggerisce naturalmente un'elaborazione mediante reti neurali ricorrenti, comunemente utilizzate nel caso di flussi di dati di lunghezza indefinita. In particolare, si è optato per l'utilizzo di due tipologie alternative di unità di elaborazione: le Gated Recurrent Unit (GRU) [14] e le Long Short Term Memory (LSTM) [29], al fine di valutarne l'efficacia nel nostro contesto e valutandone le varianti monodirezionali e bidirezionali.

Per entrambe le tipologie di reti si è scelto di utilizzare una dimensione degli stati interni pari a 256, corrispondenti alla metà della dimensione dei vettori di input, considerando come rappresentazione numerica dell'intera descrizione il valore dello stato finale oppure la media dei due stati finali nel caso bidirezionale. Tali processi di elaborazione sono rappresentati graficamente nelle figure 2.10a e 2.10b.

Approccio base per le esibizioni

Per quanto riguarda il caso dell'elaborazione delle esibizioni, come già accennato nella sezione 2.2.2, si è deciso di rappresentare ogni stanza con una serie di quattro screenshot degli interni,



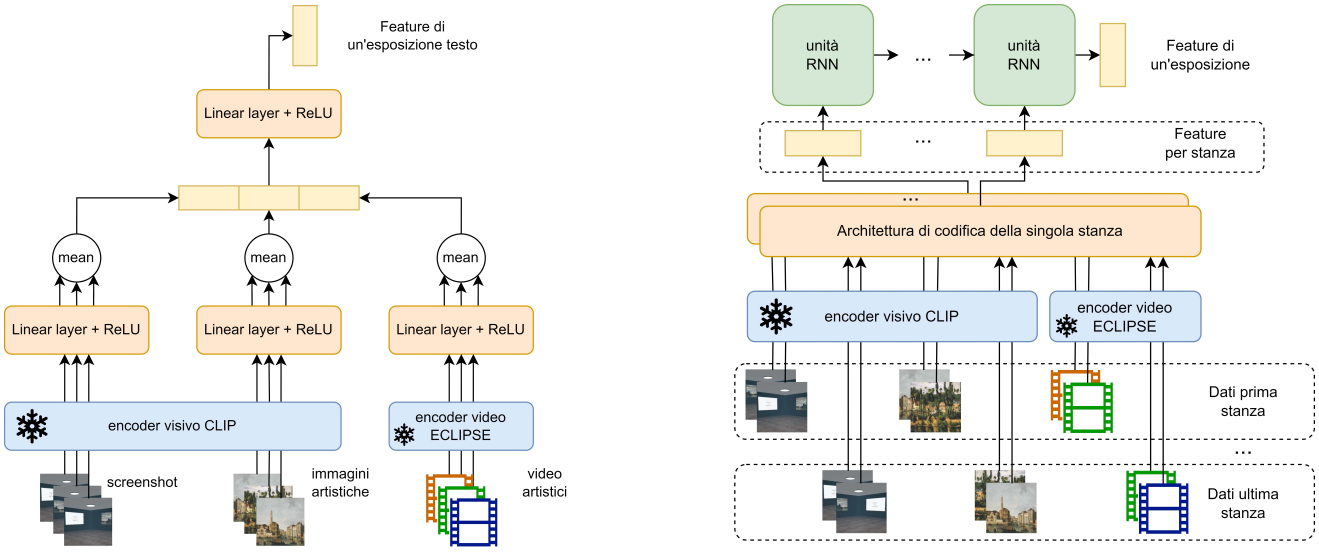
(a) RNN mono-direzionale (b) RNN bi-direzionale

Figura 2.10: Rappresentazione schematica dell'architettura utilizzate per l'elaborazione della descrizione testuale associata ad una esposizione mediante reti neurali ricorrenti. Nel caso bidirezionale si sottintende la suddivisione delle descrizioni in frasi.

andando dunque rappresentare un'intera esibizione con una sequenza di immagini di lunghezza variabile, dovuta alla presenza di diverse possibilità per il numero delle stanze. Inoltre, tale rappresentazione può essere aumentata utilizzando la serie di immagini e video delle opere esposte in ciascuna scena.

Per ottenere una rappresentazione delle immagini rappresentati gli interni, così come anche per il caso dei quadri, si è utilizzato l'encoder visivo di CLIP per eseguire l'estrazione delle relative feature. Per quanto riguarda i video, invece, si è optato per l'utilizzo alternativo di due encoder visivi appartenenti a modelli distinti: la versione base di ECLIPSE [41], analoga alla versione pre-allenata di CLIP4Clip [42], e la migliore tra le versioni di ECLIPSE [41] ottenute dopo il finetuning sul dataset video. Così facendo, si potrà in seguito andare a valutare l'eventuale impatto di tale allenamento aggiuntivo sulle performance finali. In particolare, in entrambi i casi si è adottato il formato di input migliore rispetto ai test descritti nella sezione 3.1.1, corrispondente ad una sequenza di 16 frame e altrettanti spezzoni audio da 10 secondi ciascuno, estratti in modo uniforme rispetto alla durata di ciascun video.

Una prima strategia di elaborazione, utilizzata come baseline per le successive, ha previsto l'utilizzo di una semplice rete composta da layer feed forward. In particolare, come rappresentato schematicamente in figura 2.11a, si è optato per un'elaborazione mediante un layer lineare dedicato a ciascuna tipologia di dato (screenshot, immagini artistiche e video artistici) e con funzione di attivazione ReLU. Vista quindi la natura variabile del numero di vettori ottenuti, dovuta alla variabilità nel numero di stanze così come nel numero di opere, si è optato per utilizzare una semplice operazione di mean pooling per aggregarli, ottenendo un vettore per ciascuna tipologia di feature. Per ottenere la rappresentazione finale dell'esposizione, si è utilizzato un ulteriore layer lineare privo di funzione di attivazione che trasforma la concatenazione delle feature "medie" delle diverse tipologie in un vettore della stessa dimensione delle feature testuali trasformate,



(a) Approccio base con layer lineari

(b) Approccio gerarchico

Figura 2.11: Architetture per l'elaborazione delle esposizioni d'arte multimediali aumentate dall'informazione sulle opere esposte in esse, casi base lineare e gerarchico con RNN.

corrispondente nel nostro caso a 256, come rappresentato in figura 2.11a.

In alternativa a tale strategia di elaborazione, la seconda strategia implementata si basa sull'utilizzo di layer convoluzionali. In particolare, si è scelto di interpretare il numero di feature dei vettori come il numero di canali di input, elaborando l'informazione mediante kernel monodimensionali di dimensione 3, distinti per le diverse tipologie di dato, così da permettere a ciascun elemento di essere influenzato da quelli adiacenti. Si noti tuttavia come l'ordine degli elementi dipenda solo in parte dall'organizzazione sequenziale delle stanze, presentando invece un elemento di arbitrarietà rispetto alla disposizione degli screenshot o delle opere corrispondenti a ciascuna di esse. Nella prima strategia, tale aspetto non è invece rilevante, poiché le feature di ciascun elemento sono elaborate inizialmente in modo indipendente le une dalle altre e l'operazione di mean pooling risulta invariante per permutazione.

Come nella prima strategia, si è quindi applicata una funzione di attivazione ReLU sui risultati ottenuti dall'applicazione dell'operazione di convoluzione, aggregando poi le diverse feature mediante mean pooling per ciascuna delle diverse tipologie di dato. I tre vettori finali sono quindi concatenati ed elaborati mediante l'applicazione di un layer lineare per ottenere la dimensione finale pari a 256. Tale approccio, presenta quindi una struttura analoga a quello riportato in figura 2.11a, dove vengono però sostituiti i layer lineari inferiori con quelli convoluzionali.

Approccio gerarchico per le esibizioni

Come si può notare, le precedenti architetture ignorano l'informazione sulla strutturazione delle esposizioni in stanze distinte. Per tale ragione, si è deciso di considerare anche dei modelli alternativi che tenessero in considerazione anche tale aspetto, procedendo con una elaborazione delle feature che rispettasse il rapporto gerarchico tra singolo locale e scena completa. In par-

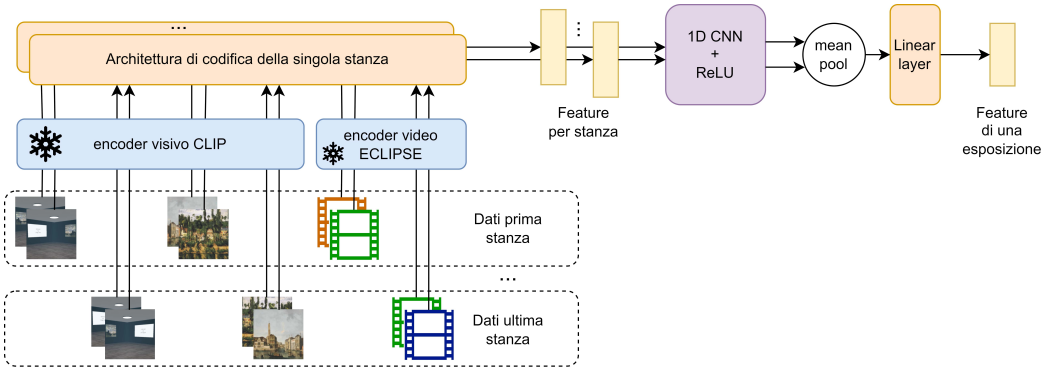


Figura 2.12: Architetture per l'elaborazione delle esposizioni d'arte multimediali aumentate dall'informazione sulle opere esposte in esse nel caso gerarchico con CNN.

ticolare, si è pensato di sfruttare lo schema di elaborazione lineare presentato precedentemente, rappresentato in figura 2.11a, per l'elaborazione delle feature appartenenti alle singole stanze, eventualmente aumentate dall'informazione sulle relative opere. Così facendo è possibile aggregare tali informazioni astraendosi dall'ordine arbitrario utilizzato per tali feature, ottenendo per ciascuna esposizione una serie di lunghezza variabile di vettori rappresentanti ciascuno una stanza decorata.

A questo punto, come nell'approccio base, sono state implementate due varianti per l'approccio gerarchico. Vista la lunghezza variabile del numero di vettori estratti per ogni museo e considerando la naturale successione di visita delle stanze, la prima strategia consiste nell'utilizzare nuovamente delle reti neurali ricorrenti. Come per il caso delle feature testuali si sono presi in considerazione unità di tipo GRU ed LSTM, così come le loro varianti monodirezionali e bidirezionali. Per chiarezza, si riporta in figura 2.11b una rappresentazione schematica dell'architettura utilizzata per questa tipologia di elaborazione gerarchica delle informazioni.

Anche nella seconda strategia l'elaborazione dei vettori delle singole stanze avviene mediante layer lineari, mentre per aggregare l'informazione sulle varie stanze viene utilizzata una rete con layer convoluzionali analoga al caso base. Come indicato anche nella sezione 2.2.2, la struttura di questa rete permette di considerare in modo congiunto l'informazione sulle stanze adiacenti. Lo schema di elaborazione di quest'architettura è riportato, per chiarezza, in figura 2.12.

Sebbene l'aspetto di adiacenza tematica delle stanze non sia presente nel dataset di esposizioni realizzate, l'idea di tale processo di elaborazione nasce dall'organizzazione delle loro controparti reali, che possono essere organizzate in sezioni tematiche formate da stanze adiacenti.

Confronto delle feature ed allenamento

Le feature appartenenti alle esibizioni virtuali e alle corrispondenti descrizioni, trasformate mediante le architetture precedentemente descritte devono quindi essere messe a confronto secondo una adeguata funzione di loss. Come già anticipato, si desidera che tale funzione permetta mediante la procedura di addestramento basato sulla discesa del gradiente, di avvicinare tra di loro

le rappresentazioni appartenenti ad elementi corrispondenti, separandole al contempo da quelle dei dati dissimili.

Come già accennato nella sezione 2.2, si è optato per l'utilizzo di una nota funzione di costo, la triplet loss [63]. Tale funzione utilizza in particolare tre elementi, chiamati *ancora*, *positivo*, e *negativo*, e mira a massimizzare la similarità tra elementi corrispondenti (ancora e positivo) minimizzando al contempo quella tra campioni dissimili (ancora e negativo). Matematicamente, la triplet loss si può esprimere come:

$$\mathcal{L}(A, P, N) = \max\left(0, s(A, N) - s(A, P) + \Delta\right) \tag{2.1}$$

dove $s(A, P)$ è la similarità tra l'ancora ed il positivo, $s(A, N)$ è la similarità tra l'ancora e il negativo, ed Δ rappresenta un margine. Quest'ultimo rappresenta più precisamente un vincolo, soddisfatto se $s(A, P) > d(A, N) + \Delta$, comportando un costo nullo, e violato altrimenti.

Nell'ambito della procedura d'addestramento seguita si procede come è prassi comune per mini-batch B composti da un insieme di stanze e testi associati, andando a calcolare la funzione di costo come:

$$\mathcal{L}_{triplet} = \frac{1}{2|B|(|B|-1)} \sum_{(s_i,t_i)\in B} \sum_{j=1, j\neq i}^{|B|} \left(\mathcal{L}(s_i, t_i, s_j) + \mathcal{L}(t_i, s_i, t_j)\right) \tag{2.2}$$

dove, in particolare, il costo per le triple viene calcolato in entrambe le direzioni, ovvero cercando di allontanare le stanze dissimili dalla coppia stanza-testo ($\mathcal{L}(s_i, t_i, s_j)$), e similmente di allontanare i testi dissimili ($\mathcal{L}(t_i, s_i, t_j)$).

Un aspetto importante da sottolineare è che sebbene tale funzione di costo sia comunemente utilizzata, si basa su un assunzione di fondo ben definita che non è necessariamente verificata negli scenari reali. Tale assunzione, così detta "instance-based", applicata al nostro scenario, prevede infatti che ciascuna descrizione sia rilevante per una ed una sola esibizione virtuale, e viceversa [75]. Non a caso, dalla formulazione si può notare come tutte le coppie di elementi non direttamente associati nel dataset vengano assunte come dissimili, prevedendo una penalizzazione nel caso risultassero eccessivamente vicini, indipendentemente dalla loro reale somiglianza. Attualmente il superamento di tale limitazione costituisce un'area di ricerca ancora attiva, trovando applicazione ad esempio nel caso del video retrieval [75, 20]. Per tale ragione, pur riconoscendo l'importanza di tale questione, si è deciso di allineare l'approccio proposto a quelli maggiormente seguiti in letteratura, lasciando l'investigazione di soluzioni che possano l'assunzione 'instance-based" a possibili sviluppi futuri.

Similmente al caso dei video, l'allenamento dei modelli si è basato sulla porzione di training del dataset, valutando le prestazioni su quella di evaluation per selezionare il modello migliore e testandolo infine sui dati di test. In particolare per tutte le architetture si sono utilizzati un batch di dimensione pari a 32 ed un learning rate pari a $1{\cdot}10^{-4}$ per un totale di 50 epoche, selezionando il

migliore modello sulla base della recall@1 sul compito di retrieval con query testuale. Per ulteriori dettagli implementativi, si rimanda alla sezione B.3 dell'appendice B.

3

Esperimenti e risultati

In questo capitolo, verranno presentate più nel dettaglio i diversi parametri di allenamento utilizzati per ciascuno dei modelli applicati per affrontare, data un'informazione testuale in input, i problemi di retrieval di video artistici e delle esposizioni. La trattazione verrà suddivisa rispetto a tali compiti presentando per ciascuno gli esperimenti svolti e i valori ottenuti rispetto a diverse metriche di valutazione. Tali risultati sperimentali verranno quindi discussi per indagare l'influenza delle differenze architetturali e metodologiche sul risultato finale.

3.1 Retrieval dei video artistici

3.1.1 Esperimenti

Come accennato nel capitolo precedente (Sezione 2.2.1), per affrontare il problema del retrieval dei video artistici sono stati utilizzati due modelli principali, CLIP4Clip [42] ed ECLIPSE [41],poi valutati sulla porzione di test del dataset completo. Dopo alcuni test preliminari per definire gli iperparametri, si è provveduto a selezionare ed eseguire i seguenti esperimenti suddivisi in base all'architettura sottostante.

Per quanto riguarda il modello basato su CLIP4Clip [42], sono stati eseguiti i seguenti test, allenando i modelli per 10 epoche con un learning rate pari a $2 \cdot 10^{-5}$ e salvando per ciascuno il modello con loss minore rispetto al validation set:

- allenamento sul dataset completo (mantenendo comunque separati le porzioni di training, validation e test), con input di 64 frame dei video e batch di dimensione 4;
- allenamento sul dataset completo, con input di 32 frame dei video e batch di dimensione 8;
- allenamento sul sottoinsieme del dataset contenente elementi con descrizioni testuali interamente processabili dall'encoder testuale (in seguito chiamato "dataset limitato"), con input di 32 frame dei video e batch di dimensione 8.

In particolare, i primi due casi sono stati selezionati per valutare il possibile impatto della dimensione dell'input, mentre l'ultimo è volto a verificare se limitare l'allenamento agli elementi la cui descrizione rientra completamente nella finestra di contesto analizzabile dall'encoder testuale potesse migliorare i risultati, compensando per la minore numerosità del dataset. Infatti, in caso di descrizioni di lunghezza maggiore, il comportamento predefinito del tokenizzatore consiste nel troncare la porzione in eccesso, utilizzando solamente l'inizio della stringa.

Per i modelli basati su ECLIPSE [41], si prendono in considerazione le seguenti casistiche, dove ciascun modello è stato allenato per 10 epoche con una dimensione del batch pari a 8:

- allenamento sul dataset completo:
  - con input pari a 8 frame video e 8 corrispondenti spezzoni audio di 10 secondi e learning rate $5 \cdot 10^{-5}$;
  - con input pari a 16 frame video e 16 corrispondenti spezzoni audio di 10 secondi e learning rate $5 \cdot 10^{-5}$;
  - con input pari a 16 frame video e 16 corrispondenti spezzoni audio di 10 secondi e learning rate $2 \cdot 10^{-4}$;
- allenamento sul dataset limitato:
  - con input pari a 8 frame video e 8 corrispondenti spezzoni audio di 10 secondi e learning rate $5 \cdot 10^{-5}$;
  - con input pari a 16 frame video e 16 corrispondenti spezzoni audio di 10 secondi e learning rate $5 \cdot 10^{-5}$;
- allenamento su dataset limitato, ma con evaluation set completo:
  - con input pari a 8 frame video e 8 corrispondenti spezzoni audio di 10 secondi e learning rate $5 \cdot 10^{-5}$;
  - con input pari a 16 frame video e 16 corrispondenti spezzoni audio di 10 secondi e learning rate $5 \cdot 10^{-5}$;
  - con input pari a 16 frame video e 16 corrispondenti spezzoni audio di 10 secondi e learning rate $2 \cdot 10^{-4}$.

Inoltre, visto che come si potrà osservare nel seguito, la maggior parte dei modelli tendeva ad ottenere le migliori prestazioni nelle prime epoche, si è deciso di aggiungere un allenamento di lunghezza inferiore per ECLIPSE nella configurazione migliore tra le precedenti, in modo da provare ad avvicinarsi in modo più graduale alla zona localmente ottimale. Più in particolare in questo caso si sono utilizzati input pari a 16 frame video, 16 corrispondenti spezzoni audio di 10 secondi e batch di dimensione 8, andando però a decrementare il learning rate a $8 \cdot 10^{-5}$ ed eseguendo l'allenamento per sole 6 epoche.

modello	dataset completo train	dataset completo eval	learning rate	epoche	dim. batch	fps/ frammenti audio	epoca migliore	vt/R1 ↑	vt/R5 ↑	vt/R10 ↑	vt/ MeanR ↓	tv/R1 ↑	tv/R5 ↑	tv/R10 ↑	tv/ MeanR ↓
CLIP4Clip	V	V	2e-05	10	4	64	1	81.82	93.18	97.73	1.86	**88.64**	97.73	97.73	1.48
CLIP4Clip	V	V	2e-05	10	8	32	2	86.36	97.73	97.73	1.50	86.36	97.73	**100.00**	1.41
CLIP4Clip	F	F	2e-05	10	8	32	4	88.64	97.73	97.73	1.61	86.36	97.73	97.73	1.45
ECLIPSE	V	V	5e-05	10	8	8	2	75.00	97.73	97.73	1.66	84.09	**100.00**	**100.00**	1.25
ECLIPSE	V	V	5e-05	10	8	16	1	84.09	95.45	95.45	1.66	**88.64**	97.73	**100.00**	1.30
ECLIPSE	V	V	2e-4	10	8	16	2	86.36	**100.00**	**100.00**	1.18	84.09	**100.00**	**100.00**	1.20
ECLIPSE	F	F	5e-05	10	8	8	3	77.27	95.45	95.45	1.84	86.36	97.73	**100.00**	1.27
ECLIPSE	F	F	5e-05	10	8	16	2	86.36	93.18	97.73	1.89	86.36	97.73	**100.00**	1.30
ECLIPSE	F	V	5e-05	10	8	8	3	77.27	95.45	95.45	1.84	86.36	97.73	**100.00**	1.27
ECLIPSE	F	V	5e-05	10	8	16	3	**90.91**	95.45	97.73	1.50	86.36	97.73	**100.00**	1.30
ECLIPSE	F	V	2e-4	10	8	16	3	79.55	97.73	**100.00**	1.41	77.27	**100.00**	**100.00**	1.32
ECLIPSE	F	V	8e-05	6	8	16	3	**90.91**	**100.00**	**100.00**	**1.16**	**88.64**	**100.00**	**100.00**	**1.18**

Tabella 3.1: Configurazioni dei diversi esperimenti di retrieval per i video artistici e relativi risultati. Per ogni riga, si evidenziano i valori migliori per le recall e il rango medio per il retrieval video-to-text (vt) e text-to-video (tv).

Per ulteriori dettagli implementativi riguardo ai agli esperimenti riguardanti CLIP4Clip[42] e ECLIPSE[41], si rimanda rispettivamente alle sezione B.2.1, B.2.2 dell'appendice B.

I risultati ottenuti dai diversi esperimenti sono riportati nella tabella 3.1, dove si evidenziano per ciascuno i valori migliori per le metriche considerate. In particolare, queste sono state divise tra quelle relative al retrieval video-to-text, dove si utilizza un video come query per recuperare la corrispondente descrizione testuale, e a quello text-to-video dove la query è costituita da una stringa e si cerca di associarvi l'opera più appropriata. Per ciascuna query, il modello produce un ordinamento degli elementi presenti nello split di test sulla base della metrica di similarità delle corrispondenti feature trasformate. Per valutare la qualità di questo ordinamento, sono state utilizzate delle metriche comunemente impiegate nell'ambito del retrieval, guardando in particolare al rango medio dell'elemento corretto ("ground truth") e alla recall@k con k pari a 1, 5 e 10. Quest'ultima metrica ha una formulazione similare alla recall classica, calcolando la frazione di volte in cui l'elemento corretto compare nei primi k risultati proposti, ossia, in formule:

$$\text{recall@k} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(\text{ground\_truth}_i \in \text{top-}k \text{ risultati per la query } i)$$

dove N rappresenta la dimensione dello split di test del dataset.

3.1.2 Discussione e confronto dei risultati

Come si può notare, in tutti i casi testati i modelli raggiungono dei valori piuttosto elevati in termini di metriche, specialmente per quanto riguarda il caso del retrieval text-to-video, di maggior interesse per l'ambito di applicazione di questo lavoro. In particolare, si può notare come per le diverse query testuali tutti i modelli riescano a inserire il video corrispondente nelle prime 5 scelte in oltre il 97% della totalità dei casi, ponendolo al primo posto nella maggior parte dei casi.

Questi risultati potrebbero essere dovuti alla numerosità piuttosto ristretta del dataset di partenza, ed in particolare alla presenza di solamente 44 elementi nell'insieme di dati utilizzati per i test. Nonostante infatti si sia cercato di prevenire l'overfitting, selezionando il modello con la loss minore rispetto ad un validation set di elementi non considerati dal modello per l'aggiornamento dei pesi, e verificando che le curve di apprendimento (learning curve) non presentassero segni di divergenza, è comunque possibile che il modello abbia basato il suo apprendimento su delle caratteristiche specifiche degli elementi presenti nel dataset o della specifica suddivisione utilizzata. In questo caso, le performance risulterebbero probabilmente significativamente ridotte se si dovesse applicare tale modello per dei video artistici provenienti da altre fonti, denotando una scarsa capacità di generalizzazione. Per verificare tale ipotesi sarebbe opportuno disporre di un insieme di opere d'arte video derivanti da fonti alternative, in modo da valutare quanto i modelli allenati possano preservare le prestazioni dimostrate nei test eseguiti. Un'altra alternativa potrebbe essere quella di verificare se dei risultati analoghi vengano raggiunti anche utilizzando delle suddivisioni diverse del dataset di partenza, adottando un approccio di cross-validation, in modo da escludere la variabile legata agli specifici split impiegati per i test precedenti.

Un'altra ipotesi è relativa alla tipologia di dati presenti all'interno del dataset. Vista infatti la numerosità ridotta, risulta più probabile che gli elementi presenti siano piuttosto diversificati, rendendo più semplice la risoluzione del compito di retrieval. In questo caso infatti i punti corrispondenti a video e descrizioni distinte potrebbero risultare già sufficientemente separati in partenza all'interno dello spazio multidimensionale delle feature.

Al di là di queste considerazioni generali, analizzando i risultati dei diversi esperimenti si può notare come utilizzare un numero di frame maggiore risulti, generalmente, in un aumento delle prestazioni relative al task del video-to-text retrieval. Un'eccezione in questo senso è costituita dal caso dei modelli basati su CLIP4Clip, dove questa tendenza appare invertita. In questo caso va però notato come la dimensione del batch sia stata ridotta da 8 a 4 per ragioni di risorse in termini di memoria video, potenzialmente impattando negativamente le capacità di generalizzazione del modello. Un comportamento analogo non sembrerebbe invece verificarsi nel caso del task inverso di text-to-video retrieval.

In entrambi i casi, si può invece apprezzare come la differenza di prestazioni tra i modelli basati su CLIP4Clip ed ECLIPSE sia limitata nonostante quest'ultimo utilizzi un numero di frame video inferiore, grazie all'aggiunta della modalità audio, favorendo alle volte i primi e altre i secondi. I risultati migliori si riscontrano comunque utilizzando l'architettura di ECLIPSE, in particolare allenando il modello unicamente sul sottoinsieme di elementi del training set con una lunghezza minore o uguale al limite massimo dell'encoder testuale, costituito da 90 dei 163 elementi totali, e monitorandone le prestazioni rispetto all'evaluation set completo, composto da 11 elementi.

L'utilizzo di tale sottoinsieme del training set, potrebbe essere vantaggioso per il modello in quanto evita l'allenamento rispetto a descrizioni testuali incomplete che potrebbero quindi

risultare in una qualità minore rispetto alle altre, catturando solo parzialmente il significato di un'opera. Un'evidenza di tale beneficio sembrerebbe essere presente guardando al caso del retrieval video-to-text dei modelli basati su ECLIPSE, dove le metriche di esperimenti allineati rispetto al learning rate e alla dimensione dell'input appaiono leggermente migliori quando l'allenamento avviene sul dataset ridotto. Un tale fenomeno non sembra invece essere presente nel caso del retrieval text-to-video.

3.2 Retrieval delle esposizioni d'arte multimediali

3.2.1 Esperimenti

Come descritto nella sezione 2.2.2, per affrontare il problema del retrieval delle esposizioni artistiche si è utilizzato un approccio di tipo feature-based, testando diverse architetture e considerando o meno l'utilizzo di feature aggiuntive relative alle opere d'arte oltre a quelle degli screenshot degli interni. In particolare, per decidere quali esperimenti eseguire, si sono considerate le combinazioni delle seguenti alternative binarie per ottenere 8 casistiche totali:

- strategia base o strategia gerarchica;
- utilizzo di layer lineari o convoluzionali per l'elaborazione delle feature iniziali (nel caso base) o delle stanze (nel caso gerarchico);
- utilizzo delle feature aggiuntive relative alle opere d'arte video e non.

Questo primo gruppo di 8 test sfruttano le feature video derivate utilizzando l'encoder del miglior modello di retrieval rispetto ai test della sezione 3.1.1, corrispondente all'ultima riga della tabella 3.1. In particolare, si è scelto di utilizzare delle GRU come tipologia di rete neurale ricorrente, bidirezionale per quanto riguarda il testo e mono-direzionale per l'elaborazione della sequenza di stanze dei musei nel caso gerarchico. Queste scelte sono motivate dal fatto che, mentre nel caso delle descrizioni testuali, in vista anche di una applicazione su query dell'utente, l'interesse è rivolto al significato della descrizione senza porre particolare interesse per l'ordine di disposizione delle frasi, nel caso delle stanze l'ordine di visita suggerito dalla disposizione rappresenta un elemento chiave che va preservato durante il processo di codifica.

Per valutare l'impatto di eventuali scelte alternative, si sono quindi considerate delle varianti alle precedenti configurazioni, definite nel seguito come "gruppi". In particolare, si sono implementate le seguenti modifiche:

- utilizzare delle LSTM al posto delle GRU (8 casi, dalla riga 8 alla riga 15);
- utilizzare come encoder per i video la corrispondente versione di ECLIPSE senza finetuning (4 casi, dalla riga 16 alla riga 19);
- utilizzare solamente GRU monodirezionali (8 casi, dalla riga 20 alla riga 27);

	LSTM	enc. desc. bidirez.	enc. scene bidirez.	gerarc.	prima elab. feature visuali	usa feature opere	enc. video feature	T2S_R@1 ↑	T2S_R@5 ↑	T2S_R@10 ↑	T2S_mean_rank ↓
0	F	V	-	F	linear	F	-	31.16 (1.85)	64.49 (3.96)	77.86 (3.13)	10.33 (1.14)
1	F	V	-	F	linear	V	EC_trained	62.91 (7.98)	86.47 (5.38)	92.65 (2.61)	3.90 (1.02)
2	F	V	-	F	conv.	F	-	54.80 (5.49)	88.55 (2.96)	95.49 (2.25)	2.91 (0.49)
3	F	V	-	F	conv.	V	EC_trained	71.26 (3.48)	95.66 (0.43)	98.58 (0.12)	1.84 (0.08)
4	F	V	F	V	linear	F	-	**97.58** (0.77)	**100.00** (0.00)	**100.00** (0.00)	**1.03** (0.01)
5	F	V	F	V	linear	V	EC_trained	93.98 (2.49)	99.92 (0.12)	**100.00** (0.00)	1.09 (0.04)
6	F	V	-	V	conv.	F	-	96.24 (1.34)	99.83 (0.12)	**100.00** (0.00)	1.05 (0.02)
7	F	V	-	V	conv.	V	EC_trained	90.39 (2.24)	99.75 (0.20)	99.83 (0.12)	1.16 (0.05)
8	V	V	-	F	linear	F	-	15.62 (1.03)	45.78 (2.40)	62.57 (3.43)	16.68 (2.53)
9	V	V	-	F	linear	V	EC_trained	18.38 (0.66)	50.29 (3.17)	66.67 (4.63)	14.76 (2.48)
10	V	V	-	F	conv.	F	-	46.45 (8.58)	81.95 (7.74)	92.82 (4.32)	4.06 (1.39)
11	V	V	-	F	conv.	V	EC_trained	53.72 (4.31)	89.22 (2.13)	95.66 (1.13)	2.92 (0.33)
12	V	V	F	V	linear	F	-	77.36 (1.54)	98.91 (0.51)	99.67 (0.31)	1.42 (0.05)
13	V	V	F	V	linear	V	EC_trained	76.52 (2.37)	98.25 (0.41)	99.67 (0.12)	1.45 (0.04)
14	V	V	-	V	conv.	F	-	66.58 (13.95)	93.48 (6.92)	97.83 (2.72)	2.15 (0.92)
15	V	V	-	V	conv.	V	EC_trained	72.01 (2.79)	95.32 (0.77)	98.25 (0.82)	1.89 (0.18)
16	F	V	-	F	linear	V	EC_base	65.91 (7.47)	89.14 (4.73)	94.07 (3.79)	3.57 (1.53)
17	F	V	-	F	conv.	V	EC_base	72.35 (7.15)	96.99 (1.34)	98.75 (0.35)	1.73 (0.20)
18	F	V	F	V	linear	V	EC_base	94.57 (1.48)	99.83 (0.12)	99.92 (0.12)	1.08 (0.03)
19	F	V	-	V	conv.	V	EC_base	88.64 (1.25)	99.42 (0.12)	99.75 (0.00)	1.21 (0.02)
20	F	F	-	F	linear	F	-	16.29 (2.21)	49.79 (1.86)	65.25 (0.43)	15.42 (1.46)
21	F	F	-	F	linear	V	EC_trained	33.42 (3.66)	61.40 (3.01)	72.60 (3.26)	12.72 (1.35)
22	F	F	-	F	conv.	F	-	26.40 (6.77)	60.74 (10.61)	73.35 (11.04)	11.68 (5.16)
23	F	F	-	F	conv.	V	EC_trained	30.49 (3.60)	66.17 (3.79)	81.37 (3.49)	8.53 (1.10)
24	F	F	F	V	linear	F	-	68.67 (4.30)	93.65 (1.90)	98.25 (0.94)	2.07 (0.35)
25	F	F	F	V	linear	V	EC_trained	62.49 (3.83)	90.81 (1.55)	96.91 (0.85)	2.64 (0.34)
26	F	F	-	V	conv.	F	-	46.03 (6.55)	79.87 (6.34)	90.06 (4.41)	4.75 (1.31)
27	F	F	-	V	conv.	V	EC_trained	48.54 (8.67)	83.29 (7.68)	91.98 (4.27)	4.02 (1.42)
28	F	V	V	V	linear	F	-	97.49 (0.20)	**100.00** (0.00)	**100.00** (0.00)	**1.03** (0.00)
29	F	V	V	V	linear	V	EC_trained	95.57 (0.66)	**100.00** (0.00)	**100.00** (0.00)	1.05 (0.01)

Tabella 3.2: Configurazioni e risultati degli esperimenti di retrieval delle esibizioni d'arte multimediale per il task text-to-scene (T2S). Per ogni metrica, è riportato il valore medio ottenuto su 3 esecuzioni, con le relative deviazioni standard tra parentesi.

- utilizzare solamente GRU bidirezionali (2 casi, righe 28 e 29).

In totale si sono quindi eseguiti 30 casi di test distinti, ciascuno allenato e testato sul dataset completo di esibizioni, utilizzando una batch size pari a 32 e un learning rate pari a $1 \cdot 10^{-4}$ per 50 epoche e selezionando alla fine il miglior modello sulla base della recall@1 per il compito di retrieval text-to-scene. Inoltre, ciascun test è stato eseguito per 3 volte, così da poterne considerare i valori medi, riportati con le relative deviazioni standard nella tabella 3.2. Per evitare un sovraccarico informativo, e considerando che l'interesse di questo lavoro risiede principalmente nel compito di text-to-scene retrieval si è deciso di riportare solamente le metriche relative a quest'ultimo. La tabella di risultati relativi al caso scene-to-text è comunque reperibile nell'appendice B alla sezione B.4 , si rimanda inoltre alla sezione B.3 per ulteriori dettagli implementativi.

3.2.2 Discussione dei risultati

Dato il numero significativo di casistiche analizzate, per rendere più chiara la discussione dei risultati, questa sezione verrà suddivisa in tre sottosezioni. Nella prima, "risultati intra-gruppo",

si analizzeranno le differenze presenti tra le diverse configurazioni presenti entro in ciascun gruppo, evidenziando eventuali parallelismi o differenze tra questi. Nella seconda, invece, si adotterà un punto di vista più di alto livello, andando a confrontare i risultati inter-gruppo , focalizzandosi sulle differenze rispetto a quello iniziale. Una sezione finale trarrà quindi delle considerazioni generali sull'insieme degli esperimenti.

Oltre a tale suddivisione, sempre nell'ottica di una maggior comprensione del lettore, si è deciso di inserire tra parentesi le righe degli esperimenti presi in esame di caso in caso, sottointendendo il riferimento alla tabella 3.2. L'unica eccezione è costituita dei casi in cui tale informazione venga piuttosto citata nel testo.

Risultati intra-gruppo

Una prima considerazione consiste nell'osservare come generalmente ci sia un netto distacco prestazionale tra i modelli base (righe 0-3,8-11,16,17,20-23) e quelli gerarchici, che sfruttano l'informazione sull'organizzazione delle esposizioni in stanze, in favore di questi ultimi. Questa differenza, seppur potenzialmente dovuta unicamente alla maggior complessità di questi ultimi modelli, suggerisce come una prima analisi delle informazioni a livello delle singole stanze possa essere benefica per il compito di retrieval.

Oltre a ciò, concentrandosi sull'insieme di esperimenti iniziali corrispondenti alle prime 8 righe, e considerando in particolare gli approcci base (0-3), si nota come l'introduzione delle informazioni aggiuntive riguardanti le opere (1 e 3) porti ad un aumento delle prestazioni dei modelli. Questo comportamento è facilmente motivabile. Infatti, gli screenshot delle stanze ritraggono solamente delle versioni compresse delle immagini artistiche originali e non possono includere l'informazione sul contenuto delle opere video a causa della loro natura statica.

Sorprendentemente però, questa situazione si ribalta nel caso degli approcci gerarchici (4-7), per i quali l'aggiunta dell'informazione sulle opere (5 e 7) va a penalizzare i modelli, specialmente rispetto al valore della recall@1. Tale comportamento parrebbe suggerire che tali architetture non stiano sfruttando queste informazioni aggiuntive o peggio che le interpretino come degli elementi di disturbo, magari a causa di un'eccessiva compressione durante le fasi di elaborazione delle feature delle singole stanze.

Questa diversa reazione dei modelli rispetto alla presenza delle informazioni aggiuntive appare generalmente consistente anche internamente agli altri gruppi di esperimenti (8-15 e 20-27), con due uniche eccezioni corrispondenti alle coppie di righe $(14, 15)$ e $(26, 27)$, entrambe basate su reti convoluzionali per l'elaborazione delle stanze. Tali osservazioni andrebbero quindi ulteriormente analizzate per verificare le ipotesi precedentemente indicate, provando ad utilizzare dei valori differenti per quanto riguarda la dimensione dei diversi layer o testando i modelli su dati alternativi.

Un'altra osservazione riguarda la differenza di prestazioni per gli approcci base a secondo dell'utilizzo o meno di layer convoluzionali. In particolare, si può notare come i modelli che

utilizzano unicamente layer lineari (0,1,4 e 5) risultino generalmente peggiori degli altri (2,3,6 e 7), anche nelle varianti dei casi di test iniziali (8-27). Questo comportamento potrebbe essere dovuto alla capacità dei layer convoluzionali di aggregare le informazioni sulle stanze ed opere più prossime, permettendo alla rete di creare delle rappresentazioni più ricche di informazioni rispetto a diverse "zone" dell'esibizione.

Passando all'analisi dei risultati relativi ai soli modelli gerarchici (4-7, 12-15, 18,19, 24-27, 28,29), si può invece notare come le prestazioni migliori siano ottenute dalle reti che uniscono le informazioni delle diverse stanze attraverso le reti neurali ricorrenti (4, 5, 12, 13, 18, 24 ,25, 28 e 29) piuttosto che sfruttare i layer convolutivi. Ad esempio nel caso degli esperimenti 24-27, i modelli 24 e 25 ottengono un guadagno di oltre 15 punti percentuali per la recall@1 rispetto alle controparti con layer convoluzionali (26 e 26).

Questo vantaggio potrebbe essere dovuto alla possibilità offerta dalle reti ricorrenti di analizzare congiuntamente l'intera sequenza degli ambienti. Le architetture basate su layer convoluzionali, al contrario, hanno una visione solamente limitata di tale sequenza, andando ad aggregare unicamente l'informazione di 3 stanze consecutive a causa della dimensione del kernel utilizzato.

Come indicato nella sezione precedente, l'idea dell'aggregazione basata su convoluzioni consiste infatti nel raggruppare le informazioni relative a possibili aree tematiche "locali" costituite da gruppi di stanze consecutive. Tale struttura però, sebbene tipicamente presente nelle esposizioni reali curate da esperti di dominio, non è stata forzata durante la creazione delle esposizioni, e richiederebbe quindi un ulteriore approfondimento su dati opportuni, per valutarne l'effettiva efficacia.

Risultati inter-gruppo

Passando ora a confrontare i risultati tra blocchi di esperimenti corrispondenti tra gli esperimenti iniziali e gli altri gruppi, si può osservare come l'utilizzo delle LSTM (8-15) al posto delle reti GRU (0-7), risulti costantemente svantaggioso per lo scenario preso in esame. Questo comportamento è probabilmente dovuto alla migliore efficienza di queste ultime, specie nel caso di dataset di dimensioni contenute e nel caso di sequenze di lunghezze ridotte [76].

Considerando ora i diversi estrattori utilizzati per le feature video, si può notare come il finetuning del modello (1,3,5,7) causi un leggero decremento delle performance rispetto agli esperimenti che utilizzano la variante base di ECLIPSE[41] (16-19). Questa tendenza potrebbe essere motivata dal fatto che l'allenamento sul compito di video retrieval va a specializzare eccessivamente l'encoder, rendendo poi più difficoltoso il suo adattamento al caso del retrieval delle intere esibizioni. Nonostante ciò, come anche in precedenza, le prestazioni dei modelli che sfruttano queste feature aggiuntive (16-19) risultano inferiori rispetto all'elaborazione dei soli screenshot (0,2,4,6).

Considerando infine le diverse scelte per quanto riguarda la direzionalità delle GRU si possono fare due considerazioni. Nel primo caso si vede come l'utilizzo di modelli monodirezionali anche

per l'informazione testuale (20-27) risulti nettamente svantaggioso rispetto al gruppo base (0-7). Questo aspetto risulta abbastanza naturale visto che nel caso del problema del retrieval si è principalmente interessati ad ottenere una rappresentazione vettoriale del testo che ne codifichi l'intera informazione semantica, senza un particolare focus sullo specifico ordine di descrizione delle opere o delle stanze. Oltre a ciò, un'elaborazione monodirezionale, potrebbe causare un eccessiva compressione delle informazioni, portando la rete a "dimenticare" il contenuto di quelle iniziali e causando quindi la creazione di feature che codificano solo parzialmente il contenuto di un'esibizione.

Prendendo invece in considerazione i due casi delle righe 28 e 29 in cui si sono utilizzate delle GRU bidirezionali anche per l'elaborazione della sequenza di stanze (caso gerarchico), si può osservare una differenza minima rispetto ai casi iniziali (4 e 5). Questa differenza rispetto alla netta differenza osservata nel caso precedente potrebbe essere dovuta al fatto che, essendo il numero di stanze piuttosto limitato (si ricorda che questo è compreso tra 6 e 9) anche l'utilizzo di una più semplice rete monodirezionale per la codifica di una serie di stanze non comporta perdita di informazione dovuta ad una potenziale sovra-compressione. Per vagliare questa ipotesi sarebbe necessario eseguire dei test su delle esibizioni più estese, e potenzialmente con un ordinamento delle opere più sofisticato in modo da poter valutare quando sia rilevante l'utilizzo di un modello in grado di codificare l'informazione sull'ordine di visita suggerito ad un visitatore.

Conclusioni generali

In conclusione, considerando globalmente i casi di test eseguiti, i risultati migliori sono stati ottenuti dall'architettura gerarchica che utilizza GRU monodirezionali per elaborare le descrizioni testuali e una combinazione di layer lineare e GRU monodirezionali per l'elaborazione delle esibizioni d'arte multimediale, senza sfruttare però l'informazione aggiuntiva sulle singole opere contenute ciascuna di esse (riga 4). Analogamente per il caso del retrieval dei video, i valori delle metriche risultano molto elevati, raggiungendo un valore di recall@1 superiore al 97%.

Sebbene in questo caso la dimensione del dataset risulti maggiore al problema precedentemente affrontato e i modelli utilizzati siano più semplici per via dell'approccio di tipo feature based, è comunque possibile che il modello sia stato in grado di apprendere dei bias inseriti involontariamente durante la generazione del dataset. Per verificare tale ipotesi, come per il caso dei video, sarebbe necessario valutare il modello su dataset alternativi possibilmente provenienti da diverse fonti.

Oltre a ciò, come notato anche nella sezione 2.2.2, ci si è posti in uno scenario che sottende l'ipotesi semplificativa del retrieval di tipo instance-based, in cui ad ogni descrizione corrisponde solamente un'esibizione e viceversa. Tale assunzione non è però completamente soddisfacente per la realizzazione un buon sistema di retrieval nel caso preso in esame. Un approccio migliore, infatti, dovrebbe prendere in considerazione anche il grado di similarità tra i diversi elementi

del dataset, producendo un ordinamento finale degli elementi più vicino all'aspettativa di un utilizzatore [75, 20].

Un ultima osservazione riguardante la validità dei risultati ottenuti, riguarda il formato delle query testuali utilizzate per eseguire la valutazione dei modelli durante l'allenamento e il test finale. Sebbene infatti questo sia la procedura standard de facto utilizzata nell'ambito del retrieval inter-modale basato su testo[57, 42, 41, 31], utilizza "delle frasi di ricerca" difficilmente analoghe a quelle che formulerebbe un utente umano. Infatti, se le descrizioni presenti nel dataset risultano estremamente verbose andando a catturare finemente molti dei dettagli riguardanti le complesse scene oggetto di questo lavoro, è ragionevole aspettarsi che un utente andrebbe ad utilizzare delle query più semplici e al contempo astratte.

Di conseguenza sarebbe opportuno andare a studiare in futuro il possibile formato di interrogazione per un tale sistema da parte di un utente, eseguendo per esempio un analisi su un campione diversificato di individui. Tale conoscenza permetterebbe quindi realizzare anche un dataset più vicino ad un reale caso l'uso per un sistema di retrieval di esibizioni virtuali, permettendo una valutazione maggiormente oggettiva delle prestazioni dei modelli presi in considerazione

Conclusioni e sviluppi futuri

In questo lavoro di tesi, si è andati a indagare l'area di ricerca recentemente nata del retrieval di spazi tridimensionali ricchi di elementi multimediali attraverso l'utilizzo di query testuali. Più in particolare si è preso in considerazione lo scenario specifico delle esibizioni virtuali d'arte multimediale, comprendenti immagini e video, andando ad affrontare il problema di retrieval sfruttando strumenti di Intelligenza Artificiale.

Vista la novità di tale scenario, ci si è dovuti scontrare con la scarsa disponibilità di dati relativi a tali esposizioni, andando quindi a creare delle procedure automatizzate per la loro generazione ex-novo. Per fare ciò, si è andati a ricercare dati ed informazioni riguardanti opere d'arte in formato digitale, rappresentate come immagini o video di natura artistica. Mentre nel primo caso ci si è potuti basare su diverse alternative già pubblicamente disponibili, nel caso dei contenuti video è stato necessario svolgere una ricerca e annotazione manuale dei diversi elementi, realizzando un dataset ad-hoc. Basandosi sui dati così ottenuti, si è quindi creato un dataset di circa 2000 esibizioni d'arte, sviluppando una procedura semi-automatica per generare gli ambienti espositivi virtuali e collocandovi le opere d'arte secondo un criterio di coerenza tematica.

Per valutare l'utilizzabilità del dataset di video collezionati e le prestazioni ottenibili su di esso in ambito di video retrieval, si è andati ad allenare e testare alcuni dei modelli pubblicamente accessibili (CLIP4Clip [42] ed ECLIPSE [41]). Tale aspetto è risultato inoltre propedeutico per la scelta del miglior modulo di analisi per tale tipologia di contenuti da poter integrare nel sistema realizzato per affrontare il caso del retrieval delle esibizioni artistiche. Tale problema è formalizzabile come retrieval di ambienti tridimensionali comprendenti elementi multimediali a partire da query testuali, derivate dalle descrizioni in linguaggio naturale delle scene e delle opere artistiche ivi contenute. Nel modellare le esposizioni artistiche, si è preso in considerazione uno scenario semplificato in cui si assume la disponibilità di sistemi di object detection ideali in grado di localizzare ed estrarre i contenuti multimediali da ciascuna esibizione. Sulla base di tale definizione, si è andati a realizzare ed allenare un sistema di Intelligenza Artificiale in grado di affrontare il problema in esame, testandone diverse configurazioni e discutendone i risultati.

I risultati sperimentali mostrano come sia possibile risolvere il problema con buone performance. In particolare, sul test set in esame, comprendente 399 esibizioni virtuali è stato ottenuto un valori di Recall@1 del 97.49%. Gli studi ablativi inerenti l'architettura utilizzata per risolvere il problema confermano inoltre i seguenti risultati. Per l'elaborazione del testo, le GRU bidirezionali hanno dato i risultati ottimali, mentre con GRU monodirezionale Recall@1 ¡ 68.67% e con LSTM bidirezionale Recall@1 ¡ 77.36%. Per l'elaborazione delle scene, l'utilizzo di una struttu-

ra gerarchica ha permesso di ottenere risultati migliori dell'uso di una struttura non gerarchica (Recall@1 ¡ 72.35%). L'uso di reti convoluzionali per effettuare l'elaborazione delle feature locali alle singole stanze ha generalmente portato a risultati leggermente peggiori rispetto ai modelli basati su reti neurali ricorrenti (Recall@1 ¡ 96.24%). L'utilizzo di feature specifiche per i video e per i quadri non ha comportato miglioramenti significativi rispetto all'uso di sole feature visive estratte a partire dagli screenshot catturati all'interno delle stanze espositive nel caso dei modelli che si basano su una struttura gerarchica.

Nonostante i buoni risultati ottenuti, va sottolineato come questo lavoro presenti diverse limitazioni e conseguenti sviluppi futuri, costituendo dunque solamente un piccolo passo verso la soluzione del problema preso in esame.

Per prima cosa la scarsa disponibilità di opere d'arte video, assieme alla mancanza di supporto di una figura esperta di dominio, ha portato alla creazione di un dataset di dimensioni piuttosto ridotte e la cui qualità delle annotazioni richiederebbe un'analisi più approfondita per valutarne l'effettivo allineamento e la somiglianza con un scenario di applicazione reale. Inoltre, i video artistici trovati durante la ricerca risultano tipicamente in rappresentazioni solamente parziali della visione artistica degli autori, mancando dell'informazione relativa al mezzo di riproduzione originale (e.g. videoproiettore, televisore a tubo catodico, ...). Tale limitazione, seppur parzialmente intrinseca al processo di digitalizzazione, potrebbe essere ridotta nel caso di una loro esibizione in uno scenario tridimensionale in cui si potrebbe cercare di ricreare quanto più fedelmente possibile l'esperienza di visione ideata dall'artista o dal curatore d'arte, sfruttando ad esempio l'utilizzo di adeguati modelli 3D per simulare i dispositivi di riproduzione originari.

In secondo luogo, sebbene il dataset delle esposizioni realizzato in questa tesi cerchi di seguire una coerenza tematica, va notato come le strategie utilizzate per organizzare le singole stanze e l'intera esposizione sono basate principalmente sulle categorie associate alle immagini utilizzate (argomento, periodo di realizzazione e stile) e sulla similarità delle descrizioni testuali. Così facendo, tali esposizioni risultano ancora ben lontane dal risultato dalla creatività e meticolosità di un reale curatore d'arte [44]. Inoltre, vi è un intrinseco disallineamento temporale tra i quadri e le opere d'arte video, le quali sono necessariamente realizzate negli ultimi sessant'anni, difficilmente colmabile a meno di integrare conoscenze di dominio fornite da esperti d'arte.

Infine, oltre agli aspetti legati ai contenuti delle esibizioni, va indicato come anche l'organizzazione spaziale utilizzata, costituita da una concatenazione lineare di stanze, costituisca una semplificazione rispetto alle reali esposizioni d'arte presenti nel metaverso, che utilizzano generalmente una struttura più libera sfruttando con maggior creatività le potenzialità offerte dal metaverso. Tale aspetto risulta strettamente correlato con la scelta della struttura dei modelli di Intelligenza Artificiale utilizzati per affrontare il problema e richiederebbe uno studio più approfondito in tal senso. Ad esempio, se si scegliesse di mantenere una suddivisione delle esposizioni in stanze, utilizzando però un più libero schema di collegamento, potrebbe risultare naturale l'utilizzo di una architettura basata sulle Graph Convolutional Network [34].

A

Dettagli sui dataset

A.1 Esempio di una descrizione di un'esibizione virtuale

Per fornire un idea più chiara delle descrizioni associate ad un esibizione si riporta nel seguito la descrizione per il museo denominato "1035_train_TYPE_landscape", focalizzandosi in particolare sulla penultima stanza.

> This art exhibition has eight rooms other than the initial lobby. [...] The seventh room contains four artworks, two of which are videos. One of the room artworks is an image artwork has the following description. This is a sombre image of violent forces of nature where the expressive feel is intensified by the fierceness of Strindberg's spatula work. He found the scene depicted at Dalarö in the Stockholm skerries in the summer of 1892. Strindberg worked very intuitively, translating his mental state into images with brief, fierce bursts of activity. That is also why he chose small-scale formats; in fact this painting, which Strindberg called The Flying Dutchman as a reference to Richard Wagner's lonely, ever roaming captain, is one of his largest. One of the room artworks is a video artwork and has the following description. Metro-Goldwyn-Mayer is the title of a 2-minute-long, 16mm colour film made by Jack Goldstein in 1975. The film shows the animated emblem of the Hollywood studio, a lions head, set in an ornamental, (almost) perfectly round frame on a ground of deepest red, flanked symmetrically by heraldically curled strips of film. Already familiar from innumerable trailers, the lion roars and jerks its head to one side. The movement is stopped at short intervals and rewound. The lion has to begin again and again without ever reaching fulfilment. The majestic gesture congeals into catatonic jerks. One of the room artworks is a video artwork and has the following description. An older man begins dancing in a public square, oblivious to the laughing gawkers who pass by. He seems to want to engage the attention of a young man sitting on a bench who begins weeping. A group of youngsters mimics and mocks the dancer. He's oblivious to them. The music is a sprightly tune that Fred Astaire might have used as he enticed

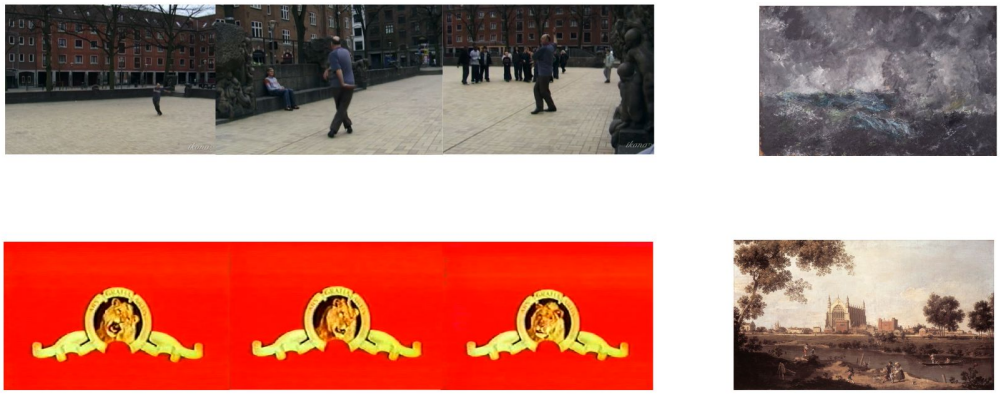


Figura A.1: Elementi multimediali presenti nella penultima stanza dell'esibizione "1035_train_TYPE_landscape".

> Ginger Rogers into his arms. The man, expressionless, continues to dance. The other man continues to weep. One kid starts taking the performance seriously and joins in the dance as a sort of pas de deux. Both men ignore him. And then it's over. One of the room artworks is an image artwork has the following description. The college and its chapel are depicted as though seen from the east, across the river Thames. A number of the buildings near to them seem to have been invented by Canaletto, and the scene as a whole, which follows a composition established in a drawing by the artist, therefore appears to be a subtle capriccio. Canaletto had visited and painted nearby Windsor Castle in 1747, and could then have made a study of the college which he later chose to integrate with other features. The view may not be an accurate record, but it is carefully composed, with the tree framing it at the left and a darkened foreground leading the eye of the viewer on into the middle-distance. The figures who fish, punt and stroll by the water effectively animate the scene. [...]

Tale stanza contiene dunque quattro d'arte di cui due video e due immagini, riportate per chiarezza nella figura A.1, ed è associata al tema paesaggistico. Come si può osservare se i quadri risultano perfettamente allineati con tale argomento, in virtù dei metadati associati agli elementi di SemArt, lo stesso non è completamente vero per i video di natura artistica. Considerando solamente questi ultimi si può infatti constatare come il primo video, intitolato "No Man Is an Island"[1] dell'artista Jesper Just, potrebbe effettivamente avere un affinità semantica con il tema dei paesaggi, ed in particolare in relazione agli ambienti urbani. A differenza di questo, la seconda opera video, realizzata dall'artista Jack Goldstein e intitolata"Metro-Goldwyn-Mayer"[2], appare invece molto più distante rispetto alla tematica dell'esibizione, aspetto che con buona probabilità è dovuto all'elemento casuale presente nella fase di creazione della configurazione per la generazione e decorazione delle scene.

[1] `https://www.ubu.com/film/just_island.html`
[2] `https://www.ubu.com/film/goldstein_mgm.html`

B

Dettagli dell'allenamento

B.1 Specifiche del sistema

Tutti i test sono stati eseguiti su un sistema con le seguenti specifiche:

- OS: Ubuntu 22.04 con kernel Linux 6.8.0
- CPU: intel code i5-4460 3.2GHz
- RAM: 4x4GB DDR3 1866MHz
- GPU: Nvidia RTX 2060 super con 8GB di VRAM
- CUDA: 12.1

Inoltre si sono utilizzate le seguenti versioni di pacchetti software:

- Anaconda: 24.5.0
- Python: 3.12.4
- pytorch: 2.3.1
- tkinter: 8.6
- BeautifulSoup: 4.12.3
- scikit-learn; 1.5.1
- Unity: 2022.3.14f1 LTS

B.2 Dettagli degli allenamenti per il retrieval dei video artistici

B.2.1 Dettagli degli allenamenti per il retrieval dei video artistici utilizzando CLIP4Clip

Oltre agli elementi indicati nella sezione 3.1.1 per quanto riguarda l'allenamento del modello CLIP4Clip[42], si sono utilizzati, analogamente al paper originale, l'ottimizzatore "BertAdam" con parametri $\beta_1 = 0.9$, $\beta_2 = 0.98$ ed $\epsilon = 1 \cdot 10^{-6}$, e uno scheduler di tipo "warmup_cosine" per aggiornare il valore del learning rate. In particolare tale funzione di aggiornamento prevede una prima riduzione lineare del learning rate durante una fase di "warmup", corrispondente nel nostro caso al 10% dell'allenamento totale, seguita da un aggiornamento più lento secondo la formula $0.5 * (1 + \cos \pi \cdot x$ dove x rappresenta il progresso del training espresso in percentuale.

B.2.2 Dettagli dell'allenamento per il retrieval dei video con ECLIP-SE

Oltre agli elementi indicati nelle sezioni 2.2.1 e 3.1.1 riguardo all'allenamento del modello ECLIPSE[41], si riportano qui alcuni dettagli aggiuntivi. In particolare, nonostante la possibilità di tale modello di gestire l'informazione audio in un paio di casi ci si è trovati con dei video unicamente visuali, privi cioè della traccia audio. In questo caso, differentemente dagli autori originali, si è deciso di fornire al modello un tensore della dimensionalità opportuna riempito di valori nulli, così da evitare di ridurre ulteriormente le dimensioni del dataset già piuttosto limitate.

Oltre a ciò, per l'allenamento si è utilizzato uno scheduler di tipo "warmup_cosine" per aggiornare il valore del learning rate analogamente al paper originale. In particolare tale funzione di aggiornamento prevede una prima riduzione lineare del learning rate durante una fase di "warmup", corrispondente nel nostro caso al 10% dell'allenamento totale, seguita da un aggiornamento più lento secondo la formula $0.5*(1+\cos \pi \cdot x$ dove x rappresenta il progresso del training espresso in percentuale.

B.3 Dettagli dell'allenamento allenamento per il retrieval delle esibizioni

Oltre ai parametri d'allenamento già indicati per il retrieval delle esibizioni nelle sezioni 2.2.2 e 3.2.1 si sono utlizzati i seguenti parametri:

- optimizer: Adam com parametri predefiniti β_1 pari a 0.9, β_2 pari a 0.999 ed ϵ pari a $1 \cdot 10^{-8}$;
- learning rate scheduler: StepLR con parametri `step_size` pari a 27 e `gamma` pari a 0.75.

In aggiunta, per i casi in cui una delle tipologie di elementi dell'input non risultasse presente, in particolare ciò si è verificato nel caso di esibizioni non contenenti alcun elemento video, e più frequentemente nel caso della strategia gerarchica in cui si lavora a livello delle singole stanze, si è scelto di utilizzare dei vettori di input nulli della dimensionalità adeguata.

Per completezza si riportano quindi le dimensionalità degli embedding relativi alle esibizioni virtuali durante l'elaborazione rispetto alle diverse tipologie di esperimenti, pari dopo l'estrazione iniziale mediante i diversi encoder a 512 per tutte le diverse modalità:

- casi base:
  - approccio basato su layer lineari: la prima elaborazione per ciascuna tipologia di feature porta la dimensionalità a 256, e analogamente il secondo layer produce un singolo output con dimensione 256;
  - approccio basato su CNN monodimensionali: si utilizzano 256 canali di uscita per l'elaborazione di ciascuna tipologia di feature e il layer lineare finale produce un output di dimensione 256;
- casi gerarchici:
  - per l'elaborazione delle feature a livello di stanza di utilizzano inizialmente dei layer lineari di dimensione 256 per ciascuna tipologia di feature, e similmente il secondo layer lineare produce un singolo output di dimensione 256;
  - per l'elaborazione delle sequenze di stanze, che riceve un input di dimensione 256 dal precedente modulo:
    - approccio basato su reti neurali ricorrenti: indipendentemente dall'utilizzo di unità di tipo GRU o LSTM, e dalla mono-direzionalità o bidirezionalità della rete, si utilizza una dimensione di output, corrispondente con la dimensione dello stato interno delle unità pari a 256;
    - approccio basato su CNN monodimensionali: si utilizzano 256 canali di uscita per l'elaborazione di ciascuna tipologia di feature e il layer lineare finale produce un output di dimensione 256.

Per quanto riguarda invece l'elaborazione delle sequenze testuali, indipendentemente dall'utilizzo di unità di tipo GRU o LSTM, e dalla mono-direzionalità o bidirezionalità della rete, si utilizza una dimensione di output, corrispondente con la dimensione dello stato interno delle unità pari a 256.

B.4 Risultati retrieval esposizioni d'arte multimediale

In tabella B.1 si riportano i risultati degli esperimenti svolti nel caso del retrieval delle esposizioni d'arte multimediale rispetto al caso sequence-to-text.

	LSTM	enc. desc. bidirez.	enc. scene bidirez.	gerarc.	prima elab. feature visuali	usa feature opere	enc. video feature	S2T_R1 ↑	S2T_R5 ↑	S2T_R10 ↑	S2T_mean_rank ↓
0	F	V	-	F	linear	F	-	27.65 (2.28)	61.07 (3.13)	76.86 (1.55)	10.78 (0.98)
1	F	V	-	F	linear	V	EC_trained	62.82 (9.69)	87.22 (5.70)	92.90 (3.33)	4.02 (1.20)
2	F	V	-	F	conv.	F	-	52.30 (6.62)	86.97 (4.29)	95.15 (2.05)	3.15 (0.62)
3	F	V	-	F	conv.	V	EC_trained	66.92 (1.75)	96.66 (0.66)	99.08 (0.24)	1.78 (0.08)
4	F	V	F	V	linear	F	-	**96.57** (0.83)	**100.00** (0.00)	**100.00** (0.00)	**1.04** (0.01)
5	F	V	F	V	linear	V	EC_trained	95.24 (1.28)	99.83 (0.24)	**100.00** (0.00)	1.07 (0.02)
6	F	V	-	V	conv.	F	-	94.07 (1.36)	99.83 (0.12)	99.92 (0.12)	1.08 (0.03)
7	F	V	-	V	conv.	V	EC_trained	89.89 (1.16)	99.67 (0.12)	**100.00** (0.00)	1.17 (0.03)
8	V	V	-	F	linear	F	-	13.87 (1.05)	42.61 (2.08)	58.90 (3.90)	18.02 (3.61)
9	V	V	-	F	linear	V	EC_trained	15.71 (0.77)	48.37 (1.14)	67.34 (0.63)	14.42 (2.85)
10	V	V	-	F	conv.	F	-	42.86 (8.33)	82.71 (6.46)	91.81 (4.31)	4.14 (1.23)
11	V	V	-	F	conv.	V	EC_trained	50.71 (6.11)	88.47 (2.66)	94.90 (1.05)	3.03 (0.40)
12	V	V	F	V	linear	F	-	75.52 (1.44)	98.33 (0.24)	99.67 (0.24)	1.49 (0.04)
13	V	V	F	V	linear	V	EC_trained	74.94 (3.25)	97.91 (0.24)	99.92 (0.12)	1.49 (0.08)
14	V	V	-	V	conv.	F	-	66.50 (12.84)	93.40 (6.68)	97.99 (2.49)	2.18 (0.97)
15	V	V	-	V	conv.	V	EC_trained	71.09 (3.72)	95.24 (1.97)	98.41 (1.05)	1.89 (0.24)
16	F	V	-	F	linear	V	EC_base	66.25 (10.05)	87.64 (4.60)	93.07 (3.49)	3.66 (1.43)
17	F	V	-	F	conv.	V	EC_base	69.26 (4.55)	96.32 (0.83)	99.16 (0.43)	1.75 (0.19)
18	F	V	F	V	linear	V	EC_base	94.32 (1.36)	99.58 (0.31)	99.92 (0.12)	1.10 (0.04)
19	F	V	-	V	conv.	V	EC_base	89.06 (0.83)	99.16 (0.24)	99.83 (0.12)	1.21 (0.02)
20	F	F	-	F	linear	F	-	15.54 (1.08)	47.12 (2.31)	64.08 (2.66)	15.62 (1.29)
21	F	F	-	F	linear	V	EC_trained	35.00 (3.97)	60.99 (4.02)	72.01 (2.07)	13.37 (0.88)
22	F	F	-	F	conv.	F	-	25.40 (6.23)	59.15 (9.08)	73.02 (10.13)	11.84 (5.18)
23	F	F	-	F	conv.	V	EC_trained	28.40 (4.02)	65.58 (3.50)	81.37 (2.73)	8.35 (1.04)
24	F	F	F	V	linear	F	-	70.93 (3.15)	94.24 (2.05)	98.08 (0.72)	1.96 (0.23)
25	F	F	F	V	linear	V	EC_trained	63.58 (1.45)	92.23 (0.54)	97.49 (0.20)	2.44 (0.18)
26	F	F	-	V	conv.	F	-	46.95 (8.30)	79.37 (5.96)	89.81 (4.71)	4.85 (1.37)
27	F	F	-	V	conv.	V	EC_trained	49.37 (9.40)	84.21 (7.62)	91.90 (5.26)	4.08 (1.46)
28	F	V	V	V	linear	F	-	95.99 (0.89)	**100.00** (0.00)	**100.00** (0.00)	**1.04** (0.01)
29	F	V	V	V	linear	V	EC_trained	96.49 (0.61)	99.92 (0.12)	**100.00** (0.00)	1.05 (0.01)

Tabella B.1: Configurazioni e risultati degli esperimenti di retrieval delle esibizioni d'arte multimediale per il caso del task sequence-to-text. I valori delle metriche sono i valori medi ottenuti su 3 esecuzioni con le relative deviazioni standard riportate tra parentesi.

Bibliografia

[1] Ali Abdari, Alex Falcon, e Giuseppe Serra. Farmare: a furniture-aware multi-task methodology for recommending apartments based on the user interests. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4293–4303, 2023.

[2] Ali Abdari, Alex Falcon, e Giuseppe Serra. Metaverse retrieval: Finding the best metaverse environment via language. In *Proceedings of the 1st International Workshop on Deep Multimodal Learning for Information Retrieval*, pp. 1–9, 2023.

[3] Ali Abdari, Alex Falcon, e Giuseppe Serra. Adoctera: Adaptive optimization constraints for improved text-guided retrieval of apartments. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*, pp. 1043–1050, 2024.

[4] Ali Abdari, Alex Falcon, e Giuseppe Serra. A language-based solution to enable metaverse retrieval. In *International Conference on Multimedia Modeling*, pp. 477–488. Springer, 2024.

[5] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Song Bai, Xiang Bai, Ngoc-Minh Bui, Minh N Do, Heyu Zhou, Yang Zhou, e altri. Shrec'18 track: 2d image-based 3d scene retrieval. *Training*, 700(70):2, 2018.

[6] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Tobias Schreck, Ngoc-Minh Bui, Trong-Le Do, Mike Holenderski, Dmitri Jarnikov, Tu-Khiem Le, Vlado Menkovski, Khac-Tuan Nguyen, Thanh-An Nguyen, Vinh-Tiep Nguyen, Van-Tu Ninh, Luis A. Pérez Rey, Minh-Triet Tran, e Tianyang Wang. Shrec'19 track: Extended 2d scene image-based 3d scene retrieval, 2019.

[7] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, e Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark, 2016.

[8] Panos Achlioptas, Maks Ovsjanikov, Kilichbek Haydarov, Mohamed Elhoseiny, e Leonidas Guibas. Artemis: Affective language for visual art. *CoRR*, abs/2101.07396, 2021.

[9] Mohamed El Ghaly Beheitt e Moez Ben Hajhmida. Automatic arabic poem generation with gpt-2, 01 2022.

[10] Best art galleries in metaverse. `https://www.metaverse-spots.com/metaverse-blog/best-art-galleries-in-metaverse`. Ultimo accesso: Settembere 2024.

[11] Lucas Beyer. Intro to transformers. `http://lucasb.eyer.be/transformer`. Ultimo accesso: Settembere 2024.

[12] Soravit Changpinyo, Piyush Sharma, Nan Ding, e Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts, 2021.

[13] Honglie Chen, Weidi Xie, Andrea Vedaldi, e Andrew Zisserman. Vggsound: A large-scale audio-visual dataset, 2020.

[14] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, e Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[15] Colette copeland sito personale. `https://colettecopeland.com`. Ultimo accesso: Settembere 2024.

[16] Diva station. `https://www.e-arhiv.org/diva/index.php?lang_pref=en`. Ultimo accesso: Settembere 2024.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, e Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[18] Emily alden foster sito personale. `https://www.emilyaldenfoster.com`. Ultimo accesso: Settembere 2024.

[19] Alex Falcon, Beatrice Portelli, Ali Abdari, Giuseppe Serra, e altri. Paving the way for personalized museums tours in the metaverse., 2024.

[20] Alex Falcon, Giuseppe Serra, e Oswald Lanz. Improving semantic video retrieval models by training with a relevance-aware online mining strategy. *Computer Vision and Image Understanding*, 245:104035, 2024.

[21] First art gallery in the metaverse. `https://firstartgallery.it/metaverse-gallery`. Ultimo accesso: Settembere 2024.

[22] Francesca fini sito personale. `https://www.francescafini.com/videoart`. Ultimo accesso: Settembere 2024.

[23] Huan Fu, Bowen Cai, Lin Gao, Lingxiao Zhang, Jiaming Wang Cao Li, Zengqi Xun, Chengyue Sun, Rongfei Jia, Binqiang Zhao, e Hao Zhang. 3d-front: 3d furnished rooms with layouts and semantics, 2021.

[24] Noa Garcia e George Vogiatzis. How to read paintings: Semantic art understanding with multi-modal retrieval. In *Proceedings of the European Conference in Computer Vision Workshops*, 2018.

[25] Yuan Gong, Yu-An Chung, e James Glass. Ast: Audio spectrogram transformer, 2021.

[26] Natalia Grincheva. Cultural diplomacy under the "digital lockdown": pandemic challenges and opportunities in museum diplomacy. *Place Branding and Public Diplomacy*, 18:8 – 11, 2021.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, e Jian Sun. Deep residual learning for image recognition, 2015.

[28] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, e Bryan Russell. Localizing moments in video with natural language, 2017.

[29] Sepp Hochreiter e Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[30] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, e Douglas Eck. Music transformer: Generating music with long-term structure. *arXiv preprint arXiv:1809.04281*, 2018.

[31] Sarah Ibrahimi, Xiaohang Sun, Pichao Wang, Amanmeet Garg, Ashutosh Sanan, e Mohamed Omar. Audio-enhanced text-to-video retrieval using text-conditioned feature alignment, 2023.

[32] jdownloader official website. `https://jdownloader.org/`. Ultimo accesso: Settembere 2024.

[33] Kadist. `https://kadist.org/paris/`. Ultimo accesso: Settembere 2024.

[34] Thomas N. Kipf e Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[35] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, e Juan Carlos Niebles. Dense-captioning events in videos, 2017.

[36] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, e Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016.

[37] Jie Lei, Tamara L. Berg, e Mohit Bansal. Qvhighlights: Detecting moments and highlights in videos via natural language queries, 2021.

[38] Fengling Li, Lei Zhu, Tianshi Wang, Jingjing Li, Zheng Zhang, e Heng Tao Shen. Cross-modal retrieval: A systematic review of methods and future directions, 2023.

[39] Junnan Li, Dongxu Li, Caiming Xiong, e Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, e Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[41] Yan-Bo Lin, Jie Lei, Mohit Bansal, e Gedas Bertasius. Eclipse: Efficient long-range video retrieval using sight and sound, 2022.

[42] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, e Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval, 2021.

[43] Magmart 100x100=900. `https://www.magmart.it/900/`. Ultimo accesso: Settembere 2024.

[44] Freda Matassa. *Organizing Exhibitions: a handbook for museums, libraries and archives*. facet publishing, 2014.

[45] Kabir Matwala, Taner Shakir, Chetan Bhan, e Manish Chand. The surgical metaverse. *Cirugía Española*, 102:S61–S65, 2024.

[46] Statista metaverse - worldwide. `https://www.statista.com/outlook/amo/metaverse/worldwide`. Ultimo accesso: Settembere 2024.

[47] Osservatorio Extended Reality & Metaverse. Cos'è il metaverso e come funziona, esempi e tecnologie. `https://blog.osservatori.net/metaverso-cos-e-come-funziona-esempi-tecnologie#:~:text=Il%20Metaverso%20rappresenta%20un%20ecosistema,accedendo%20anche%20tramite%20dispositivi%20immersivi.` Ultimo accesso: Settembere 2024.

[48] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, e Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. *CoRR*, abs/1906.03327, 2019.

[49] Youssef Mohamed, Mohamed Abdelfattah, Shyma Alhuwaider, Feifan Li, Xiangliang Zhang, Kenneth Ward Church, e Mohamed Elhoseiny. Artelingo: A million emotion annotations of wikiart with emphasis on diversity over language and culture. *arXiv preprint arXiv:2211.10780*, 2022.

[50] Youssef Mohamed, Faizan Farooq Khan, Kilichbek Haydarov, e Mohamed Elhoseiny. It is okay to not be okay: Overcoming emotional bias in affective image captioning by contrastive

data collection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume abs/2204.07660, 2022.

[51] National Gallery of Arts open data. `https://www.nga.gov/open-access-images/open-data.html`. Ultimo accesso: Settembere 2024.

[52] Ora kolmanovsky sito personale. `https://www.ofmuzar.com/newsite_2011/motion.php`. Ultimo accesso: Settembere 2024.

[53] John Ousterhout. tkinter documentation. `https://docs.python.org/3/library/tkinter.html`. Ultimo accesso: Settembere 2024.

[54] Heracles Papatheodorou. Github ubu24h. `https://github.com/Arty2/ubu24h`. Ultimo accesso: Settembere 2024.

[55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, e E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[56] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, e Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, 2016.

[57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, e Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[58] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, e altri. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[59] Joseph Redmon, Santosh Divvala, Ross Girshick, e Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[60] Shaoqing Ren, Kaiming He, Ross Girshick, e Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[61] Leonard Richardson. Beautiful soup documentation. Ultimo accesso: Settembere 2024.

[62] Babak Saleh e Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature, 2015.

[63] Florian Schroff, Dmitry Kalenichenko, e James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[64] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, e Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs, 2021.

[65] Shahar marcus sito personale. `https://shaharmarcus.com`. Ultimo accesso: Settembere 2024.

[66] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, e Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding, 2016.

[67] Silvia de gennaro sito personale. `https://www.assaus.it/degennaro`. Ultimo accesso: Settembere 2024.

[68] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Massimiliano Corsini, e Rita Cucchiara. Artpedia: A New Visual-Semantic Dataset with Visual and Contextual Sentences. In *Proceedings of the International Conference on Image Analysis and Processing*, 2019.

[69] Brick Project Studio. Unity asset store apartment kit. `https://assetstore.unity.com/packages/3d/environments/apartment-kit-124055`. Ultimo accesso: Settembere 2024.

[70] Joseph Tu. Meetings in the metaverse: Exploring online meeting spaces through meaningful interactions in gather. town, 2022.

[71] Ubuweb. `https://www.ubu.com/film/index.html`. Ultimo accesso: Settembere 2024.

[72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, e Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[73] Wikipedia performance art. `https://it.wikipedia.org/wiki/Videoarte`. Ultimo accesso: Settembere 2024.

[74] Wikipedia videoarte. `https://it.wikipedia.org/wiki/Performance_art`. Ultimo accesso: Settembere 2024.

[75] Michael Wray, Hazel Doughty, e Dima Damen. On semantic similarity in video retrieval, 2021.

[76] Shudong Yang, Xueying Yu, e Ying Zhou. Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. In *2020 International workshop on electronic communication and artificial intelligence (IWECAI)*, pp. 98–101. IEEE, 2020.

[77] GitHub yt-dlp. `https://github.com/yt-dlp/yt-dlp`. Ultimo accesso: Settembere 2024.

[78] Fuyang Yu, Zhen Wang, Dongyuan Li, Peide Zhu, Xiaohui Liang, Xiaochuan Wang, e Manabu Okumura. Towards cross-modal point cloud retrieval for indoor scenes. In *MultiMedia Modeling*, pp. 89–102, Cham, 2024. Springer Nature Switzerland.

[79] Juefei Yuan, Hameed Abdul-Rashid, Bo Li, e Yijuan Lu. Sketch/image-based 3d scene retrieval: Benchmark, algorithm, evaluation. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 264–269, 2019.

[80] Kun Zhou, Fadratul Hafinaz Hassan, e Gan Keng Hoon. The state of the art for cross-modal retrieval: A survey. *IEEE Access*, 11:138568–138589, 2023.

[81] Luowei Zhou, Chenliang Xu, e Jason J. Corso. Towards automatic learning of procedures from web instructional videos, 2017.