



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

Gestione di video in streaming on demand in un contesto di realtà virtuale

RELATORE

Prof. Fabio Palomba

Università degli studi di Salerno

CANDIDATO

Giacinto Adinolfi

Matricola: 0512107764

*A mio nonno,
per avermi insegnato ad amare gli altri ma soprattutto me stesso.*

Sommario

L'attuale aumento dell'interesse per la Realtà Virtuale (VR) è avvenuto con la disponibilità di prodotti VR commerciali di base, come gli Head Mounted Display (HMD), sviluppati da Oculus e altri fornitori, con prestazioni sempre migliori a un prezzo accessibile al grande pubblico. Le applicazioni in ambito di realtà virtuale utilizzano questi display che, indossati sulla testa, implementano capacità stereoscopiche per offrire un'esperienza di immersione totale. Per accellerare l'adozione dei dispositivi VR da parte degli utenti, i fornitori di contenuti dovrebbero concentrarsi sulla produzione di contenuti immersivi di alta qualità, per questi dispositivi. In questo studio cerchiamo di capire diversi aspetti relativi alla rappresentazione di contenuti in streaming video on demand all'interno di dispositivi di realtà virtuale, allo scopo di contribuire a stabilire le conoscenze di base relative all'implementazione di un sistema di streaming video totalmente virtuale. Svilupperemo una applicazione per Oculus Quest 2, basandoci sul kit di integrazione del visore di Oculus, con il motore grafico Unity, in modo da fornire una esperienza completamente immersiva in un contesto di realtà virtuale che dia la possibilità di visionare ed interagire con dei contenuti in streaming on demand.

Indice

Indice	ii
Elenco delle figure	iv
Elenco delle tabelle	vi
1 Introduzione	1
1.1 Contesto applicativo	1
1.2 Motivazioni ed obiettivi	2
1.3 Risultati	3
1.4 Struttura della tesi	3
2 Stato dell'arte	4
2.1 Realtà estesa	4
2.1.1 Concetto di realtà	4
2.1.2 Extended Reality (XR)	4
2.1.3 Realtà aumentata (AR)	5
2.1.4 Realtà virtuale (VR)	7
2.1.5 Realtà mista (MR)	7
2.1.6 Dispositivi XR	8
2.1.7 Head-Mounted Display (HMD)	8
2.1.8 Periferiche di input	9
2.1.9 Dispositivi VR	11

2.1.10 Dispositivi AR e MR	14
2.1.11 Domini applicativi	17
2.2 Streaming video e audio	19
2.2.1 Concetto di streaming	19
2.2.2 Meccanismo di funzionamento dello streaming	21
2.2.3 Streaming on demand	22
2.2.4 Streaming in live	25
2.2.5 Quali servizi di streaming scegliere	27
2.2.6 Lo streaming pirata	28
2.3 Concetti di intelligenza artificiale, Machine Learning e Deep Learning	29
2.3.1 Intelligenza artificiale (AI)	29
2.3.2 Machine Learning, Deep Learning e Reti Neurali	30
2.4 Algoritmi di AI applicati agli streaming video	33
2.4.1 Streaming video classification using Machine Learning	33
2.4.2 A Deep Learning Model for Extracting Live Streaming Video Highlights using Audience Messages	36
2.4.3 DeSVQ: Deep Learning Based Streaming Video QoE Estimation	39
3 Applicazione sviluppata	43
3.1 Obiettivi e descrizione dell'applicazione	43
3.2 Specifiche tecniche dell'applicazione	46
3.3 Strumenti e tecnologie utilizzate	56
3.4 Architettura dell'applicazione	58
3.5 Valutazione preliminale dell'applicazione	61
4 Conclusioni	63
4.1 Sviluppi futuri	63
4.1.1 Google-Speech-To-Text	63
4.1.2 DialogFlow	64
4.1.3 Estrapolazione dati in tempo reale	65
4.1.4 Introduzione di algoritmi di intelligenza artificiale	65
4.2 Conclusioni	66
Ringraziamenti	67

Elenco delle figure

1.1	Campo di utilizzo di Oculus Quest All-in-one	2
2.1	Reality-Virtuality Continuum introdotto da Paul Milgram e Fumio Kishino (1994)	5
2.2	Modulo di realtà aumentata dell'applicazione di IKEA	6
2.3	Esempio di realtà aumentata basata sulla sovrapposizione	6
2.4	Alcuni dispositivi di XR sul mercato	8
2.5	Controller VR di Oculus Quest 2	9
2.6	Esempio di periferiche input VR Glove	10
2.7	Esempio di periferica basata su hand tracking	10
2.8	Visori Oculus, da sinistra : Oculus Rift, Oculus Go, Oculus Quest, Oculus Rift-S, Oculus Quest 2	12
2.9	Alcuni dei visori HTC Vive, da sinistra : Vive Focus, Vive Pro, Vive Cosmos .	12
2.10	Dispositivi di supporto prodotti da Google per l'utilizzo dello smartphone come HMD	13
2.11	Visori VR della Lenovo	13
2.12	Alcuni dei visori VR prodotti da Samsung	13
2.13	Playstation VR	13
2.14	Amazon Echo Frames	15
2.15	Google Glass Enterprise Edition 2	15
2.16	Moverio BT-40	15
2.17	Microsoft Hololens 2	16

2.18 Vuzix Blade Upgraded Smart Glasses	16
2.19 Lenovo ThinkReality A3	17
2.20 Esempio di AR applicato all'ambito logistico	18
2.21 Comuni piattaforme di straming video utilizzate	21
2.22 Streamer in live sulla piattaforma Twitch	26
2.23 Software Napster odierno	28
2.24 Schema di una semplice rete neurale	32
2.25 Struttura della rete neurale di partenza	34
2.26 Esempio di estrazione di Highlight	38
2.27 Architettura del modello	39
 3.1 Stanza in cui viene generato l'utente	44
3.2 Stranza rappresentante la cucina, facilmente raggiungibile dall'utente	44
3.3 Progress bar e barra di interazione del video player	45
3.4 Interazione col video player, lato utente	46
3.5 Direction light	54
3.6 Spotlight	54
3.7 Reflection probe	55
3.8 Skybox scelta per la scena	56
3.9 Estensione utilizzata in Visual Studio Code	57
3.10 Rendering, in Blender, dell'oggetto telecomando	58
3.11 Architettura dell'applicazione sviluppata	58

Elenco delle tabelle

2.1	Accuratezza della previsione per ogni servizio	35
2.2	Tabella di accuratezza finale	36
2.3	Caratteristiche dei video utilizzati per la valutazione del modello	38
2.4	Risultati in comparazione	39
2.5	Correlazione tra le metriche oggettive e i punteggi di QoE calcolati per fotogramma	40
2.6	Confronto delle prestazioni del modello DeSVQ sul dataset LIVE Netflix I con i modelli QoE esistenti	41
2.7	Confronto delle prestazioni del modello DeSVQ sul database LIVE NFLX II con i modelli QoE esistenti	41
2.8	Confronto delle prestazioni del modello DeSVQ su Mobile Stall II con i modelli QoE esistenti	42

CAPITOLO 1

Introduzione

1.1 Contesto applicativo

La realtà virtuale (VR) sta suscitando sempre più interesse. In passato la realtà virtuale (VR) veniva studiata solo nei laboratori universitari o nei centri di ricerca, con alcuni tentativi di commercializzazione non andati a buon fine. La fornitura di apparecchiature di VR ai clienti finali era ostacolata da numerose sfide. In passato, ad esempio, si presentavano problemi di scomodità dovuti alle grandi dimensioni delle cuffie, alla scarsa qualità degli schermi, alla mancanza di contenuti creati appositamente per i dispositivi di VR e alla scarsa precisione del tracciamento della testa. Nel corso del tempo, i ricercatori e l'industria hanno cercato di trovare soluzioni a questi problemi. Prima dell'uscita di Oculus Rift, non è stato realizzato nulla di significativo. In seguito, molte delle principali aziende del settore informatico hanno rilasciato i propri dispositivi virtuali.

Ad oggi, si sta dando molta importanza al concetto di streaming di contenuti per i dispositivi di realtà virtuale. Come Netflix, i principali servizi di streaming video, tra cui anche YouTube, consentono attualmente lo streaming di video per i dispositivi di VR cercando di garantire una esperienza di utilizzo completamente immersiva. Per vedere i contenuti in VR, gli utenti utilizzano in genere gli Head Mounted Display (HMD), come l'Oculus Rift. Tramite l'ausilio di questi dispositivi, gli utenti possono muovere la testa all'interno dell'area immersiva in tutte le direzioni possibili, come si può vedere nella figura 1.1.



Figura 1.1: Campo di utilizzo di Oculus Quest All-in-one

I dispositivi più moderni consentono di tracciare un'area di gioco all'interno della quale potersi muovere. Solitamente, su questi dispositivi sono montate delle videocamere che entrano in azione quando si esce al di fuori dell'area virtuale di gioco, dando all'utente la possibilità ristabilire il contatto con la realtà.

1.2 Motivazioni ed obiettivi

Il progetto di tesi nasce dalla curiosità di capire come poter lavorare con flussi di video, in streaming on demand, all'interno di un ambiente virtuale, cercando di comprenderne le criticità e i processi produttivi che si affrontano nello sviluppo di un'applicazione del genere. È anche previsto uno spazio dedicato all'analisi di sistemi basati su intelligenza artificiale (AI), applicati a flussi di video in streaming, allo scopo di capire come poter interagirvi ed estrapolare informazioni, come:

- Le scene migliori di un video in live streaming.
- Informazioni sulla qualità di esperienza di un flusso in streaming.
- Classificazione dei pacchetti di dati, provenienti da un video in streaming, come appartenenti ad uno dei servizi sotto analisi.

In questo lavoro di tesi è previsto anche lo sviluppo di una applicazione utilizzando il software Unity, un motore grafico molto valido, che permette la creazione di videogiochi in 2D, 3D, VR,

AR, mobile e molto altro. In seguito, ci concentreremo sulla comprensione dei principi alla base della VR e dello streaming e racconteremo i dettagli implementativi dell'applicazione. Questa prevede lo sviluppo di una scena in realtà virtuale che consente l'interazione con svariati oggetti della scena e con un video player, utile nella riproduzione di contenuti in streaming on demand.

1.3 Risultati

È stata approfondita la conoscenza relativa al mondo della realtà virtuale, degli streaming video e degli algoritmi di intelligenza artificiale (AI) applicati agli streaming video. È stato possibile mettere in relazione il mondo della realtà virtuale con quello degli streaming video, creando una applicazione di medio livello che implementa tutte le componenti di base, essenziali per permettere la visione di flussi di video in un ambiente completamente immersivo. Sono stati studiati possibili sviluppi futuri capaci di migliorare l'attuale applicazione, integrando algoritmi di intelligenza artificiale o API capaci di introdurre componenti di dialogo all'interno dell'applicazione.

1.4 Struttura della tesi

La tesi è articolata in quattro capitoli:

1. Nel primo capitolo è presente l'introduzione relativa al lavoro di tesi sviluppato.
2. Nel secondo capitolo è collocato lo stato dell'arte. In questo capitolo analizzeremo i concetti di realtà estesa, i vari dispositivi utilizzati in questo ambito e i domini applicativi della realtà virtuale e aumentata. Analizziamo poi il concetto di streaming, il meccanismo di funzionamento e le varie tipologie di streaming. Il capitolo si chiude con qualche concetto relativo all'intelligenza artificiale, Machine Learning, Deep Learning e analisi di algoritmi di AI applicati agli streaming video.
3. Nel terzo capitolo parleremo dell'applicazione sviluppata: obiettivi, specifiche tecniche, tool utilizzati e architettura dell'applicazione.
4. Il quarto capitolo chiude il lavoro di tesi e contiene una breve valutazione preliminare dell'applicazione sviluppata, vengono descritti eventuali sviluppi futuri e vengono fatte le considerazioni conclusive sul progetto di tesi.

CAPITOLO 2

Stato dell'arte

2.1 Realtà estesa

2.1.1 Concetto di realtà

C'è sempre un po' di incertezza su ciò che davvero vuole significare il concetto di realtà. Per anni filosofi e letterati ne hanno cercato una definizione, potendo solo dedurre la soggettività di tale argomento. Ognuno vive una propria realtà e la costruisce sulla base delle proprie percezioni, esperienze ed emozioni. La realtà quindi si può definire come sinonimo di quelle esperienze vissute nella nostra individualità, portandoci all'inganno proprio perché rappresenta la nostra percezione. Nonostante sia magari condivisa, non coincide con ciò che gli altri stanno percependo.

2.1.2 Extended Reality (XR)

L'Extended Reality, o abbreviata con XR, è un termine che si riferisce allo studio e sviluppo di una realtà immersiva composta dall'intero spettro dal "completamente reale" al "completamente virtuale" nel concetto di *Reality-Virtuality Continuum*[1]. Essa è, infatti, una estensione del mondo reale ottenuta grazie a nuovi metodi di interazione che permettono di percepire la realtà circostante in modo differente e potenziarla. Il *Reality-Virtuality Continuum*, definito da Paul Milgram, è una scala che abbraccia tutte le possibili composizioni di oggetti

reali e virtuali. Nell'immagine sottostante, è possibile capire meglio la suddivisione di questo spettro.

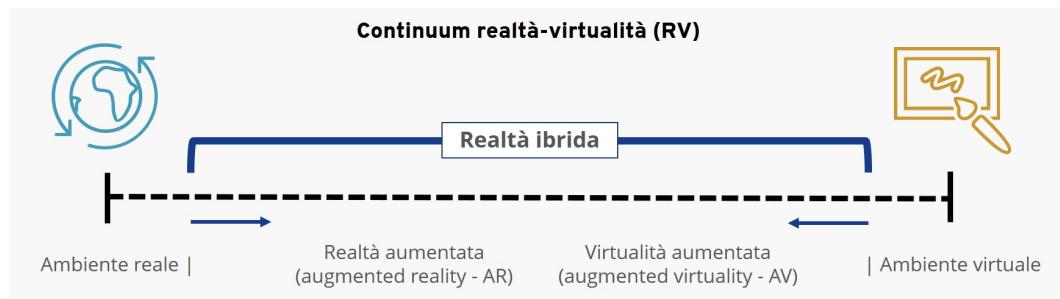


Figura 2.1: Reality-Virtuality Continuum introdotto da Paul Milgram e Fumio Kishino (1994)

L'area tra i due estremi, in cui si mescolano sia il reale che la realtà virtuale (VR), è chiamata realtà mista (MR). Molti sostengono che questo a sua volta contenga sia la realtà aumentata (AR) dove il virtuale aumenta il reale, sia la virtualità aumentata (AV) dove il reale aumenta il virtuale. L'XR è un campo in rapida crescita che viene applicato in una vasta gamma di ambiti come intrattenimento, marketing, formazione, smart working, e così via... Attualmente ci sono 3 forme di XR : Realtà Virtuale, Realtà Aumentata e Realtà Mista.

2.1.3 Realtà aumentata (AR)

La realtà aumentata[2] è una tecnologia avanzata basata sulla visione potenziata della realtà attraverso l'uso di elementi virtuali combinati con il mondo reale. Uno dei suoi punti di forza è la poca potenza di calcolo necessaria poiché è accessibile attraverso un qualsiasi smartphone, tablet o PC provvisto di camera. L'obiettivo dell'AR è quello di puntare alla percezione audiovisiva dell'utente, arricchendola con immagini, testi, suoni e modelli 3D in sovrapposizione con ciò che vede (concetto di *overlay*). La natura stessa dell'AR prevede che il sistema determini lo stato del mondo reale per poi trasferire le caratteristiche nel mondo virtuale, facendo percepire gli elementi virtuali come facenti parte della realtà.

Esistono principalmente 4 sistemi di realtà aumentata:

1. **AR basata su marker:** è basata sul riconoscimento di immagini. È una tecnologia che funziona grazie a specifici software in grado di riconoscere, tramite l'ausilio della fotocamera del proprio dispositivo, disegni in bianco e nero (come i QR Code), in modo da rilevare l'oggetto e fornire ulteriori informazioni, come: l'avvio di un servizio, l'apertura di una pagina web, la riproduzione di un video o la visualizzazione di immagini in 3D.

2. **AR basata sulla posizione:** è anche chiamata “realtà aumentata basata sulla posizione”.

È l'unico tipo di tecnologia AR che non utilizza alcun sistema di riconoscimento ma sfrutta i rilevatori di posizione ed orientamento presenti sugli smartphone come GPS, bussola digitale, misuratore di velocità e accelerometro. L'obiettivo di questa tecnologia è consentire all'utente di individuare attività commerciali nelle vicinanze, mappare direzioni ed itinerari da percorrere e tutte le altre possibili applicazioni incentrate sulla geolocalizzazione.

3. **AR proiettata:** permette di visualizzare oggetti virtuali sulla superficie del mondo reale e renderlo ancora più interattivo con l'aiuto di sensori. Un esempio è l'applicazione di IKEA grazie alla quale poter proiettare oggetti di arredamento all'interno della stanza per avere una prima impressione dell'oggetto di cui si è interessati.

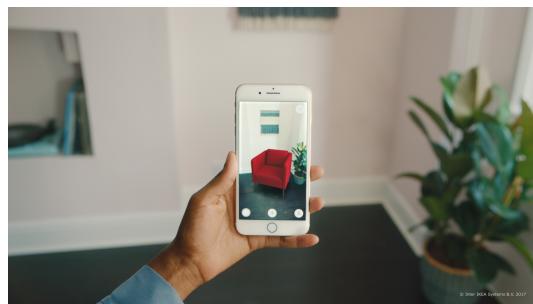


Figura 2.2: Modulo di realtà aumentata dell'applicazione di IKEA

4. **AR basata sulla sovrapposizione:** uno dei più importanti tipi di realtà aumentata poiché in grado di sostituire parzialmente o completamente la vista originale di un oggetto con una vista aumentata dello stesso. Il marketing sfrutta tanto questa tecnologia in quanto, ad esempio, un potenziale cliente può divertirsi a valutare e provare diversi prodotti grazie alla realtà aumentata basata sulla sovrapposizione di oggetti virtuali sul mondo reale.

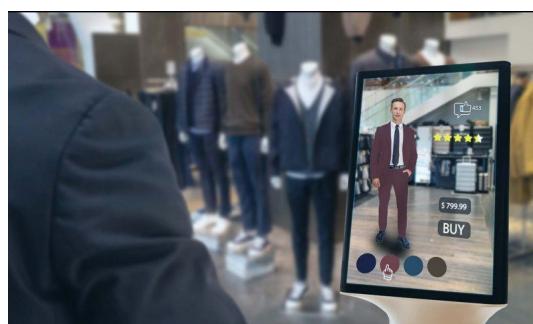


Figura 2.3: Esempio di realtà aumentata basata sulla sovrapposizione

2.1.4 Realtà virtuale (VR)

La Realtà Virtuale (VR)[3] rappresenta un'avanzata interfaccia uomo-computer che simula un ambiente realistico. È una tecnologia che dà agli utenti l'impressione di essere completamente immersi in un ambiente virtuale, creato digitalmente, con il quale possono comunicare utilizzando particolari periferiche. Il termine "interattività" descrive la capacità di interagire con gli eventi del mondo virtuale. La realtà virtuale alimenta una vasta gamma di campi. Si tratta più di una fusione di discipline precedentemente distinte che di un nuovo ramo tecnologico. Cibernetica, progettazione di database, sistemi in tempo reale e distribuiti, simulazione, grafica computerizzata, ingegneria umana, stereoscopia, anatomia umana e persino vita artificiale sono tutti campi di applicazione della realtà virtuale. Un sistema di realtà virtuale deve avere tre caratteristiche:

1. Risposta alle azioni dell'utente.
2. Grafica 3D in tempo reale.
3. Dare all'utente un senso di immersione.

Il mondo virtuale assume molte forme diverse, ad esempio:

- **La simulazione in cabina:** utilizzata per l'addestramento dei piloti delle compagnie aeree.
- **La realtà simulata:** utilizza la tecnologia di proiezione per ottenere un sistema che eguali la qualità degli schermi delle postazioni di lavoro in termini di risoluzione, colore e sfarfallio.
- **Telepresenza:** grazie alla tecnologia di telepresenza, gli utenti possono manipolare oggetti nel mondo virtuale e vedere i risultati in una postazione remota nel mondo reale.
- **Realtà virtuale desktop:** utilizza input e output convenzionali e sensori di dati a forma di guanto che interpretano il movimento delle dita, catturando così le azioni impartite dall'utente.

2.1.5 Realtà mista (MR)

La realtà mista [4] si trova in sovrapposizione tra la realtà aumentata e la realtà virtuale. Differisce dalla realtà aumentata poiché si elaborano le immagini provenienti dal mondo

reale con elementi virtuali ancorati ad oggetti reali. Questi, poi, saranno resi componenti capaci di interagire tra di loro e l’utente, fornendo non solo una realtà arricchita come l’AR, ma una vera e propria esperienza immersiva. Gli elementi virtuali vengono rappresentati da ologrammi, trattati come veri e propri oggetti integrati nel mondo reale. L’ologramma riconosce il mondo e sa fornire la proprietà di occlusione: un ostacolo si frappone tra l’utente e l’ologramma.

Le ultime periferiche create, ci consentono di interagire con gli ologrammi semplicemente toccandoli, attraverso una *Natural Interface*. Questo termine introduce un tipo di interazione che gli esseri umani sono già predisposti ad utilizzare, in quanto naturale: si può, infatti, interagire con questi ologrammi come con gli oggetti fisici del mondo reale.

2.1.6 Dispositivi XR

Nel parlare dei dispositivi XR, si fa sempre confusione sul definire l’appartenenza di un dispositivo a questa categoria, confusione purtroppo alimentata dall’utilizzo spropositato del termine XR. Come affermato nei capitoli precedenti, l’Extended Reality rappresenta tutte quelle branche che hanno a che fare con la manipolazione della realtà, ma ognuna di queste realtà avrà bisogno di un suo proprio sistema di input.



Figura 2.4: Alcuni dispositivi di XR sul mercato

2.1.7 Head-Mounted Display (HMD)

Lo sviluppo di dispositivi che agiscono nella branca della realtà estesa, definisce una distinzione di genere tra prestazione e immersione rivolta all’esperienza utente. Nonostante ciò, i dispositivi mostrano una progettazione di base comune dovuta all’utilizzo di un Head-Mounted Display (HMD). Un tipico HMD, è composto da 1 o 2 piccoli display (uno per ogni

occhio) integrati con lenti o specchi semitrasparenti all'interno della struttura di supporto del dispositivo. Il loro intento è quello di emulare la visione stereoscopica dell'occhio umano, dando l'illusione di profondità nello spazio. In base all'ambito di utilizzo, la tecnica di proiezione delle immagini potrebbe variare: ad esempio, per la realtà virtuale i display mostrano una scena di un mondo virtuale, per la realtà aumentata vengono combinate le immagini prodotte digitalmente con quello che l'utente percepisce visivamente dal mondo reale.

2.1.8 Periferiche di input

Per consentire l'interazione dell'utente con la applicazione XR vengono utilizzate delle periferiche di input di vario genere:

- **Controller VR:** hanno forme diverse, ognuna delle quali cerca di ottenere un'ergonomia quanto più naturale possibile. Ogni tasto di input è provvisto di sensori di prossimità, utilizzati a livello software per comporre animazioni procedurali nel tentativo di rappresentare coerentemente la posizione della mano anche nel mondo virtuale.



Figura 2.5: Controller VR di Oculus Quest 2

- **Gloves VR & Haptic Suit:** alcuni studi hanno permesso di sviluppare periferiche più avanzate ed ergonomiche con possibilità di restituire un feedback aptico o tattile. Queste hanno le sembianze di veri e propri guanti (figura 2.6) che, attraverso un sistema di motorini cablati, sono capaci di tracciare singolarmente le dita dell'utente. Purtroppo, lo sviluppo di questo tipo di periferiche è attualmente prematuro, poiché non è stata ancora trovata una soluzione per ridurre la latenza di tracciamento o l'eccessiva dimensione della periferica, che ne aumenta i costi di produzione. Lo stesso criterio di ragionamento potrebbe essere applicato alle tute aptiche: vere e proprie periferiche che

vengono indossate, e tramite sensori, accelerometri e giroscopi tracciano il movimento del corpo intero. Alcuni prototipi più avanzati presentano dei motori aptici, in zone ben precise, per simulare l'impatto di un oggetto contro l'avatar dell'utente nel mondo virtuale.



Figura 2.6: Esempio di periferiche input VR Glove

- **Hand Tracking:** altri produttori di dispositivi XR hanno, da alcuni anni, avviato uno studio su tecniche ed algoritmi per il tracciamento in tempo reale delle mani. Il tracciamento avviene grazie ad una semplice camera, consentendo di aumentare il livello di immersione di un'esperienza VR (figura 2.7). I comandi input vengono interpretati tramite gestures. I brand attualmente impegnati in questo campo sono l'HTC, Meta, UltraLeap e OpenXR (libreria open-source per AR-VR), insieme a tante altre. Questa tipologia di input sta diventando un nuovo standard per le esperienze immersive.



Figura 2.7: Esempio di periferica basata su hand tracking

2.1.9 Dispositivi VR

I dispositivi per la realtà virtuale sono entrati nell'immaginario collettivo come dei "caschetti" (detti anche visori) al cui interno sono presenti 2 display HMD. Quest'ultimi possono essere regolati in base al valore IPD dell'utente (*Internal Pupillary Distance - Distanza Interpupillare*). I visori sono provvisti di sensori, per il tracciamento di movimento della testa, che consentono di ricreare quello stesso spostamento nel mondo virtuale. I sensori possono variare dal giroscopio all'accelerometro o un sistema di luci strutturato (infrarossi). Questa stessa struttura viene replicata anche per le mani dell'utente, attraverso dei controllers che consentono l'emulazione di semplici interazioni quotidiane quali ad esempio la presa, il puntamento, e così via... Attualmente sono in sviluppo anche periferiche ad-hoc per il tracciamento di oggetti reali all'interno del mondo virtuale, dette Tracker.

I visori possono essere classificati in 3 categorie principali:

1. **Tethered**: la periferica è cablata ad un Desktop PC o Console per poter sfruttare la sua potenza di calcolo.
2. **Standalone**: la periferica ha un proprio centro di elaborazione incorporato. Fornisce maggiore libertà di movimento ma una resa visiva minore.
3. **Standalone Smartphone**: l'HMD viene emulato da uno smartphone che posto all'interno di un supporto si converte in una periferica VR. Le limitazioni sono quelle di un qualsiasi smartphone.

Per quanto complessa possa essere l'idea dietro la progettazione del visore, uno dei problemi maggiormente presente riguarda l'utente: si parla del *motion sickness*, ovvero quella sensazione di nausea, mal di testa e sconfinamento che si prova nelle prime interazioni con la realtà virtuale. Lato software, sono state introdotte alcune tecniche per ridurre questi sintomi concentrando un maggior effort attorno al comfort dell'utente. Il motion sickness è molto soggettivo e dipende spesso dalle caratteristiche tecniche del visore. Ad esempio, il gap temporale tra il tracciamento della testa e la risposta visiva sul display deve essere un tempo compreso tra i 7-15 millisecondi accompagnato da una frequenza di aggiornamento almeno superiore ai 90 Hz. Questo ci porta ad uno dei vari vincoli a cui la realtà virtuale è sottoposta:

- **Latenza**: i requisiti di latenza sono significativamente più alti di quelli di un videogioco standard. La GPU ha bisogno di abbastanza potenza per poter renderizzare la giusta quantità di frame nel minor tempo possibile. Una soluzione è il *Foveated Rendering*

che consiste nel ridurre la qualità dell'immagine nelle aree periferiche dei display, favorendo un rendering più veloce senza perdere di dettaglio nella zona di focus.

- **Qualità e risoluzione del display:** la chiarezza dell'immagine dipende molto dalla risoluzione e dalla qualità delle ottiche in uso nel visore. Nelle prime versioni, in cui la risoluzione era particolarmente limitata, si aveva a che fare con il problema dello Screen-door Effect: l'utente percepiva lo spazio fisico tra le righe e le colonne dei pixel nel display.
- **Lenti:** responsabili di mappare il display su un campo di visione più ampio. Forniscono anche un punto di focus più comodo per l'utente. Le lenti di Fresnel sono una lavorazione particolare che suddivide la singola lente in tante sezioni a favore del foveated rendering. Con le lenti possono avvenire distorsioni o aberrazioni di colore, corrette successivamente via software.

Da all'incirca 30 anni, il numero di visori VR presenti sul mercato, è cresciuto incredibilmente portando una varietà di dispositivi non indifferente. Di seguito ne vengono citati alcuni:

- **Oculus VR Family:** prodotti da Meta.



Figura 2.8: Visori Oculus, da sinistra : Oculus Rift, Oculus Go, Oculus Quest, Oculus Rift-S, Oculus Quest 2

- **HTC Vive VR Family:** prodotti da HTC.



Figura 2.9: Alcuni dei visori HTC Vive, da sinistra : Vive Focus, Vive Pro, Vive Cosmos

- **Google Cardboard & Daydream View:** prodotti da Google. Possono essere utilizzati con il proprio smartphone.



Figura 2.10: Dispositivi di supporto prodotti da Google per l'utilizzo dello smartphone come HMD

- **Lenovo Mirage Solo & Lenovo Explorer:** prodotti da Lenovo.



Figura 2.11: Visori VR della Lenovo

- **Samsung Gear VR & Odyssey+:** prodotti da Samsung.



Figura 2.12: Alcuni dei visori VR prodotti da Samsung

- **Playstation VR:** prodotto dalla Sony e utilizzabile soltanto su console PS.



Figura 2.13: Playstation VR

2.1.10 Dispositivi AR e MR

La sperimentazione dell’ambiente aumentato avviene con l’ausilio di speciali dispositivi di visualizzazione, che sono generalmente indossati sul corpo, chiamati *Wearable Device*. Il loro utilizzo non è solo limitato alla realtà aumentata, ma anche alla realtà mista.

I dispositivi si possono classificare in 2 categorie:

1. **Optical See Through:** l’immagine viene proiettata attraverso un display semi-trasparente permettendo l’osservazione concorrente del mondo reale.
2. **Video See Through:** utilizza una telecamera che cattura l’immagine reale e aggiunge elettronicamente le immagini virtuali per aumentare la realtà.

Le prime iterazioni di questa tecnologia hanno beneficiato dei bassi requisiti tecnici; infatti, l’AR è disponibile attraverso un qualsiasi dispositivo dotato di camera. Queste prime versioni sono limitate dal fatto che il device deve essere retto attraverso le mani, vincolando buona parte dell’esperienza utente, non a caso l’AR era inizialmente considerata una tecnologia di overlay, dove gli elementi virtuali venivano combinati in sovrapposizione al mondo reale.

Le iterazioni successive hanno portato la progettazione di questi dispositivi a focalizzarsi sulle possibilità fornite all’utente durante l’interazione. Lo sviluppo di algoritmi per l’Hand Tracking ha solo favorito una profondità di immersione maggiore. Si è passati da semplici dispositivi che sovrapponevano elementi virtuali, ad avanzatissimi sistemi che mappano il mondo reale generando una copia virtuale. Da dispositivi che vengono tenuti fra le mani a degli ergonomici occhiali che proiettano le immagini direttamente davanti agli occhi dell’utente, consentendo l’interazione tramite le gesture delle mani.

È qui che viene definita una separazione tra dispositivi AR e MR. I dispositivi per la realtà mista, sono un’evoluzione di quelli AR: non solo aumentano la realtà, ma ne consentono un’interazione che si estende nel mondo reale. Numerosi sono i dispositivi che agiscono nel campo della realtà aumentata e mista:

- **Amazon – echo frame:** dispositivo che ha bisogno di pair con uno smartphone Android, per essere poi utilizzato per la ricezione di notifiche, messaggi e chiamate. Consente, inoltre, di integrare i servizi di Amazon Alexa all’interno del dispositivo AR.



Figura 2.14: Amazon Echo Frames

- **Google Glass:** primo wearable device per l'AR che purtroppo non ha avuto il successo sperato, causa costo altissimo e vari problemi tecnici e legati alla privacy.
- **Google Glass Enterprise Edition 2:** evoluzione dei Google Glass, implementati attraverso sistema operativo Android.



Figura 2.15: Google Glass Enterprise Edition 2

- **Moverio:** prodotti dalla Epson. Principalmente per il settore di manutenzione e assistenza remota. Utilizzati in alcuni casi anche per formazione specialistica.



Figura 2.16: Moverio BT-40

- **Hololens 1 & 2:** prodotti dalla Microsoft. Sono i dispositivi per la realtà mista attualmente più sofisticati del mercato. Hololens 2 è ancora in fase di sviluppo e nonostante sia un prototipo, costa all'incirca 3000 dollari. Non accessibile al grande pubblico.



Figura 2.17: Microsoft Hololens 2

- **Vuzix Smart Glass:** dispositivi AR che hanno trovato posto in diversi settori: specialistico e non specialistico. Consentono un utilizzo anche quotidiano per un pubblico più casual.



Figura 2.18: Vuzix Blade Upgraded Smart Glasses

- **ThinkReality:** prodotti dalla Lenovo. Sono dispositivi per la realtà mista, hanno la potenza di un portatile e permettono di usufruire di una visualizzazione 3D grazie alle lenti basate sulle tecnologie dei monitor stereoscopici.



Figura 2.19: Lenovo ThinkReality A3

2.1.11 Domini applicativi

Ad oggi le tecnologie immersive sono un campo in continua crescita. La realtà estesa viene vista come un nuovo strumento di interazione nella società. Di seguito parliamo di alcuni degli ambiti maggiormente influenzati dalle realtà immersive:

- **Industria:** l'AR e il VR sono ormai parte dell'industria digitale del futuro, che vede continue innovazioni come '*Additive & Smart Manufacturing* (Stampa 3D), *l'Internet of things*, *Artificial Intelligence*, *Industria 4.0* fino alle nanotecnologie. Le tecnologie immersive stanno svolgendo un ruolo positivo nell'intero settore industriale globale.

I casi di impiego dell'AR sono molteplici, a supporto di quasi tutte le attività che si svolgono all'interno degli stabilimenti industriali. È possibile intervenire nelle fasi di produzione, in un contesto di totale sicurezza per il personale e consentire il monitoraggio di tutti i processi di assemblaggio e costruzione di un prodotto.

Un esempio sono i tecnici della Porsche che utilizzano dispositivi AR per proiettare comunicazioni e grafici, consentendo agli esperti da remoto di fornire un supporto in tempo reale.

L'utilizzo della realtà estesa riduce significativamente anche i costi delle operazioni di stoccaggio. Alcune aziende stanno già testando sistemi di realtà aumentata mobili che consentono il riconoscimento degli oggetti in tempo reale tramite codice a barre e navigazione interna (figura 2.20). Indossando uno dei wearable devices, si possono visionare le informazioni necessarie direttamente sul campo visivo dell'utente.



Figura 2.20: Esempio di AR applicato all’ambito logistico

- **Medicina:** i campi di applicazione della realtà virtuale in ambito medico riguardano principalmente riabilitazione motoria/cognitiva, terapia di disturbi psichiatrici e apprendimento tramite simulazioni. Le simulazioni procedurali e chirurgiche basate sul VR stanno rivoluzionando la formazione in campo medico aiutando a fornire competenze cliniche coerenti con il percorso formativo.

Le innovazioni nel campo AR possono aiutare a migliorare la capacità di medici e chirurghi nel diagnosticare, trattare ed eseguire interventi chirurgici. Il grande vantaggio, infatti, è quello di poter fornire, direttamente davanti gli occhi del medico, le informazioni sul paziente o sull’intervento in corso.

- **Ingegneria civile:** nello specifico, nell’ambito dell’architettura e delle costruzioni, la visualizzazione di un’ambiente per mezzo di una ricostruzione tridimensionale è molto più comprensibile ed immediata rispetto ai tradizionali planimetrie. La possibilità di vagliare ipotesi di progettazione in tempo reale, prima ancora che queste vengano rese nella realtà. Le applicazioni in questo campo non si limitano solo a questo: durante i lavori è possibile usufruire della realtà aumentata per illustrare gli schemi tecnici riguardanti tubature e sistemi elettrici.
- **Automotive:** dopo le prime esperienze in ambito pubblicitario e promozionale, il settore automotive scopre nuove possibilità grazie alle realtà immersive. Alcuni dei grandi marchi hanno già iniziato a sperimentare diverse applicazioni: da informazioni dell’auto proiettate sul cruscotto (AR) a simulatori di guida (VR). In alcuni autosalone, l’acquirente ha potuto customizzare un’auto in tempo reale attraverso l’AR. Le vere applicazioni di queste tecnologie si spostano su l’assistenza remota, con degli smart glass che mostrano le informazioni dell’auto in tempo reale.

- **Intrattenimento:** questo settore sfrutta maggiormente le tecnologie immersive (nello specifico AR, VR). Le prime applicazioni di intrattenimento si riconducono alle esperienze multisensoriali dei parchi divertimento (stanze con schermi a 360° e piattaforme moventi). Ad oggi, esistono svariate applicazioni AR-VR orientate al gaming: uno dei più grandi successi AR fu Pokémon GO che, utilizzava l'AR in concorrenza con il tracciamento GPS, per spingere l'utente a camminare per le strade alla ricerca dei Pokémon.

Molto apprezzate sono anche le applicazioni di Streaming Video live e on demand in VR, che garantiscono un' esperienza immersiva nettamente superiore rispetto all'utilizzo di Smartphone, Tablet o Personal Computer.

2.2 Streaming video e audio

2.2.1 Concetto di streaming

Dopo anni di miglioramenti tecnologici, si è potuto, finalmente, introdurre lo streaming multimediale [5], tempo fa limitato da tecnologie poco evolute e vincoli di rete.

Lo streaming è un modo di guardare o ascoltare contenuti online, senza l'obbligo di doverli scaricare sul dispositivo da cui si sta effettuando richiesta. Basta solamente selezionare il contenuto di interesse e, dopo pochi istanti, il questo viene riprodotto.

Chi ha vissuto l'epoca del noleggio dei film in VHS ricorda quanto fosse faticoso e macchinoso poter accedere ai contenuti video. Bisognava infatti uscire di casa e recarsi nel negozio di videonoleggio alla ricerca del film che si voleva vedere, e sperare che questo fosse disponibile. Con il passaggio dalle VHS ai DVD, sono arrivate le prime forme di interattività e contenuti extra, ma il vero passo in avanti è stato fatto con l'avvento di internet e delle infrastrutture necessarie.

Se tempo fa per poter guardare un contenuto bisognava attendere che questo fosse mandato in onda in televisione, con l'avvento dello streaming questo non è più necessario. Lo streaming, infatti, si è diffuso rapidamente grazie al progressivo e rapido miglioramento dei dispositivi hardware che possediamo in casa: Personal Computer, Smart TV, Tablet, smartphone, che hanno integrato schermi con sempre più alta risoluzione e memorie sempre più ampie. Un altro fattore che ha consentito la rapida espansione degli streaming è la connessione internet. Anche questa si è rapidamente evoluta, passando da una velocità di

56kbps, negli anni 90, ad una velocità odierna di circa 1Gbps grazie alle attuali connessioni in fibra ottica.

I media, l'industria e l'istruzione stanno assistendo a uno sconvolgimento significativo grazie allo streaming. Su computer, smartphone e altri dispositivi, milioni di persone guardano in diretta notizie, sport e concerti. Sia le aziende che le università utilizzano la tecnologia dello streaming per incrementare la condivisione delle informazioni, migliorare la cooperazione aziendale e condividere programmi di apprendimento a distanza. Le società di media e le emittenti stanno aprendo i loro archivi, dando ai consumatori l'accesso a migliaia di film popolari e a lungo dimenticati (documentari e 50 anni di programmazione televisiva).

Ad oggi, infatti, possiamo godere di contenuti in streaming di alta qualità, grazie all'avvento di numerose piattaforme di streaming, quali Netflix, Amazon Prime, Spotify e molte altre.

Secondo il rapporto Global Entertainment & Media Outlook 2022-2026 di PwC [6], i contenuti OTT (Over-the-top) sono cresciuti del 35,4%, nel pieno della pandemia di Covid-19 del 2020, e sono aumentati del 22,8% nel 2021, con ricavi pari a 79,1 miliardi di dollari. Secondo PwC, anche se con ritmo più moderato, i ricavi legati all'OTT aumenteranno entro il 2026, con un CAGR pari al +7,6% e ricavi pari a 114 miliardi di dollari.

Uno dei vantaggi dello streaming è quello della totale libertà che si ha nella visualizzazione del contenuto. Si può, infatti, decidere di mettere in pausa il contenuto, retrocedere/avanzare la riproduzione di qualche secondo o anche cambiare titolo/traccia qualora quello scelto non si rivelasse di gradimento.

Lo streaming è molto utilizzato anche per l'ascolto di musica, in questo caso si parla di streaming audio. Tempo fa, infatti, c'era bisogno di un CD o una chiavetta USB per poter essere capaci di ascoltare della musica, ed erano anche periferiche non modificabili o complesse da modificare. Ad oggi invece basta avere un smartphone (o gli altri dispositivi sopra citati) e una connessione ad internet, per poter ascoltare in maniera, spesso gratuita, contenuti audio di qualsiasi tipo, in streaming.



Figura 2.21: Comuni piattaforme di straming video utilizzate

2.2.2 Meccanismo di funzionamento dello streaming

Per essere riprodotto in streaming, il contenuto online viene diviso in pacchetti di dati. Questi pacchetti di dati vengono inviati al browser che si sta utilizzando, che immagazzina un numero di pacchetti dopo il quale comincia a riprodurre il contenuto. La riproduzione avviene attraverso un lettore audio o video, che raccoglie i pacchetti di dati in un buffer (contenitore di dati), il quale, una volta pieno, manda i dati in output all’utente. Nel caso si abbia una connessione Internet troppo lenta, potrebbero esserci problemi di buffering. In queste situazioni bisogna attendere qualche secondo prima che nel buffer si sia accumulata una quantità di dati sufficienti per poter ricominciare la riproduzione.

Aziende grandi come Netflix hanno bisogno di server e piattaforme in Cloud per l’archiviazione dei contenuti da visualizzare. Dispongono anche di reti per la distribuzione dei contenuti che hanno il compito di mantenere, in una cache e in prossimità del luogo in cui verranno trasmessi, i contenuti maggiormente riprodotti in modo da ridurre la latenza e facilitare la fruizione del contenuto.

Analizzando il lato relativo al consumatore del contenuto, gli unici requisiti sono quelli di avere un abbonamento ad una determinata piattaforma, quando serve, e una connessione accettabile; in genere si ha bisogno di una velocità pari a 2 Mbps per una buona esperienza di streaming e almeno 5 Mbps per una esperienza in full HD o 4K.

La fruizione dello streaming può avvenire in due modi: streaming on demand e streaming in live. Analizziamo, di seguito, le differenze tra le due tipologie di streaming.

2.2.3 Streaming on demand

Con il termine streaming on demand (letteralmente “flusso su richiesta”) [7] si identificano tutti i servizi disponibili solo su richiesta, senza un preciso palinsesto o un orario. Si contrappone ai programmi televisivi tradizionali, che hanno un orario preciso dove viene trasmesso il contenuto.

I file audio e video fruibili vengono caricati direttamente su un server in modalità compressa e sono a disposizione, degli utenti, in una finestra temporale che si ritenga utile alla fruizione. Così facendo, i file saranno a disposizione degli utenti in qualsiasi momento successivo alla registrazione dell’evento. Questi flussi possono essere accessibili a tutti gli utenti in maniera gratuita o stipulando un abbonamento con le aziende fornitrice dei servizi di streaming.

Il vantaggio della scelta di uno streaming on demand sta nel fatto che l’utente, ovunque si trovi e a qualsiasi ora del giorno, abbia sempre accessibilità al flusso e lo possa guardare quante volte voglia, può rivederlo e gestirlo come più gli piace.

Un altro vantaggio si ha per gli utenti che non hanno disponibilità di una buona connessione internet per poter guardare un determinato evento in live streaming; infatti, gli eventi in diretta vengono registrati e caricati sulle dovute piattaforme per renderli fruibili agli utenti in qualsiasi momento. Citiamo alcune piattaforme molto utilizzate per la fruizione di video in streaming on demand:

- **Netflix:** è indubbiamente il colosso del momento. La selezione di contenuti è molto variegata, con una costanza di aggiornamento settimanale. Il punto di forza di Netflix è la produzione di serie TV e contenuti proprietari, come: *Orange Is the New Black*, *La regina degli scacchi*, *The Umbrella Academy* e molti altri. L’abbonamento premium consente di poter guardare fino a 4 stream simultaneamente e consente l’accesso a contenuti in 4k, a differenza dell’abbonamento standard che consente di poter visualizzare fino a 2 stream contemporaneamente e con qualità fino alla HD. Entrambe le versioni permettono di poter scaricare dei contenuti sul proprio dispositivo ed usufruirne offline.
- **Amazon Prime Video:** è la piattaforma streaming di Amazon, a cui si può accedere gratuitamente se si ha un abbonamento al servizio Prime di Amazon o pagando una piccola quota mensile, nel caso non lo si abbia. Amazon Prime Video è una ottima alternativa a Netflix, ha un catalogo video molto variegato con una buona costanza di aggiornamento. Anche Amazon Prime Video si sta concentrando sulla creazione di Serie Tv e titoli in esclusiva molto validi, come: *LOL: Chi ride è fuori*, *Celebrity Hunted*:

Caccia all'uomo, *The Boys* e molte altre. Anche Amazon Prime Video offre contenuti fino ad una risoluzione di 4K.

- **Disney+:** è il servizio di streaming di *The Walt Disney Company*, arrivato in Italia nel 2020. Esso contiene un catalogo di contenuti appartenenti alla Disney, Marvel, LucasFilm (Star Wars), Pixar e altri. È quello che più si ispira a Netflix sia come estetica dell'applicazione che nell'ambito della gestione degli account. Disney+ possiede contenuti un po' per tutta la famiglia e possiede una *Kids Mode*, una modalità pensata appositamente per bambini, con una interfaccia e dei contenuti dedicati. Anche Disney+ permette di poter guardare contenuti in alta risoluzione (4k in Dolby Vision) e consente di poter scaricare e guardare contenuti offline.
- **Discovery+:** è la piattaforma di streaming di Discovery, disponibile anche su Amazon Prime Video Channels. Discovery+ sostituisce Dplay+, nella quale erano disponibili, on demand, i programmi in onda sui canali di Discovery. Il catalogo offre una vasta gamma di programmi non fiction, intrattenimento real life e sport. Anche in questo caso c'è bisogno di sostenere un abbonamento mensile, in base alle tipologie di servizio.
- **YouTube:** è una delle piattaforme di streaming più utilizzata. A differenza delle altre, YouTube è una piattaforma completamente gratuita il cui unico limite è la presenza di banner pubblicitari nell'intermezzo della riproduzione del contenuto on demand. Esiste comunque una versione a pagamento per eliminare eventuali pubblicità durante la riproduzione del contenuto. YouTube si distacca dalla tipologia di contenuti caricati dagli altri sistemi, poiché i contenuti su YouTube possono essere caricati da qualsiasi persona dotata di una telecamera e qualcosa da raccontare. Col passare degli anni YouTube si migliora sempre di più, implementando sottotitoli, per ogni lingua, ai contenuti e inserendo diversi formati di video (come i video a 360 gradi o in 3D). Da pochi anni c'è anche la possibilità di potersi abbonare a determinati canali, pagando una quota mensile, per poter guardare contenuti esclusivi, dando la possibilità ai creator di poter investire più denaro nelle attrezzature e creare contenuti migliori. I video di YouTube possono anche essere incorporati all'interno di un sito web di qualsiasi tipo, tramite una copia del codice del video all'interno del sito che si sta creando.
- **Chili:** è un servizio che si differenzia dagli altri poiché offre delle differenti metodologie di visione di uno stream. Infatti, tratta il concetto del pagare solo ciò che si guarda. Il film che si vuole guardare può essere noleggiato o acquistato, evitando di dover stipulare un

abbonamento mensile. Con il noleggio ci sono 28 giorni di tempo per decidere quando visionare il contenuto. Dal primo “play” si hanno 48 ore per completare la visione del contenuto o per rivederlo. Con l’acquisto invece si può guardare il contenuto senza alcun limite.

Citiamo alcune piattaforme molto utilizzate per lo streaming audio on demand:

- **Spotify:** è un servizio di musica in digitale che consente l’ascolto in streaming di milioni di brani su dispositivi differenti. È tra i servizi di streaming audio più utilizzati e più conosciuti. Spotify contiene un gran catalogo di brani e una vasta gamma di funzionalità che consentono agli utenti di scoprire nuova musica, ascoltare playlist automatiche divise per genere, ascoltare podcast e altro ancora. È indubbiamente il servizio che offre la più ampia compatibilità di piattaforma. Spotify ha una versione gratuita che permette, da computer, di poter ascoltare brani con un numero limitato di skip. La riproduzione viene spesso interrotta da pubblicità (circa ogni 15 minuti di ascolto) e la qualità audio è limitata. Da app le cose cambiano, poiché non è possibile scegliere quali brani ascoltare ma le tracce vengono riprodotte casualmente. Con la versione a pagamento si ha accesso a tutte le funzionalità dell’applicazione, senza interruzioni pubblicitarie e i brani possono essere scaricati per un ascolto in offline e con una risoluzione audio migliore.
- **Apple Music:** rappresenta indubbiamente il miglior servizio di streaming audio per gli utenti Apple. Il suo catalogo prevede circa 75 milioni di brani. Non ha una versione gratuita ma prevede un abbonamento mensile. Apple Music consente un ascolto dei brani in qualità molto alta grazie al formato lossless e per alcuni brani consente l’audio spaziale Dolby Atmos. C’è però un dettaglio non trascurabile, l’iPhone o gli auricolari forniti dalla Apple non supportano una risoluzione audio di tale qualità, per questo bisogna utilizzare, per un ascolto al massimo delle prestazioni, un DAC o un paio di cuffie cablate.
- **Qobuz:** è uno dei migliori servizi di streaming musicale per gli audiofili, grazie alla sua compatibilità con vari sistemi audio. Il suo catalogo prevede circa 70 milioni di brani che possono essere anche acquistati tramite la boutique digitale Qobuz. Ha una versione gratuita che permette solo l’ascolto dell’anteprima del brano. Tra i punti di forza di Qobuz c’è sicuramente l’ottima app realizzata, che permette un utilizzo molto

intuitivo e una interfaccia piacevole. Permette anche l’ascolto su computer tramite browser web.

- **Tidal:** è il miglior servizio di streaming musicale per gli audiofili. Non offre però una versione gratuita. È il principale competitor di Qobuz ma ha, come vantaggio, il fatto di avere un catalogo più ampio di quello di Qobuz. Come svantaggio invece c’è il fatto che la versione HiFi ha un costo elevato rispetto a Qobuz, che ha anche una più ampia compatibilità coi sistemi HiFi.
- **Amazon Music Unlimited:** è l’app di streaming musicale di Amazon. Rappresenta un’ottima alternativa a Spotify o Apple Music. Anche Amazon Music prevede un abbonamento mensile ma ha dei vantaggi per chi fa parte dell’ecosistema Amazon. Infatti, oltre ai classici abbonamenti ad un vasto catalogo di musica, permette anche agli utenti abbonati ad Amazon Prime, accesso gratuito ad Amazon Prime Music. Questo però da accesso ad un catalogo molto meno vasto rispetto a quello della versione Unlimited. Inoltre, esiste anche la versione gratuita, Amazon Music Free, che consente l’ascolto di playlist, ma con inserzioni pubblicitari.
- **YouTube Music:** è l’app di streaming musicale di Google, compresa nell’abbonamento YouTube Premium. Ha un buon catalogo di brani ma non ha una qualità audio molto alta, per questo è consigliato ad utenti che fruiscono di una grande quantità di video in streaming on demand e vogliono, in un solo colpo, liberarsi degli spot pubblicitari ma avere anche accesso a un catalogo di streaming musicale. Google sta comunque continuando ad investire per migliorare il servizio di musica.

2.2.4 Streaming in live

Con il termine live streaming [7] si intende la trasmissione di contenuti, via internet, in tempo reale. Negli ultimi tempi questo concetto ha preso piede grazie ai numerosi progressi dei dispositivi hardware e della sempre maggiore affidabilità e velocità delle connessioni ad internet.

Sul Web vengono trasmessi in diretta spettacoli, concerti, conferenze, convegni, corsi di formazione, lezioni, eventi sportivi e aziendali, eventi formativi o medicali e altri ancora (figura 2.22). Anche le dirette sui social, come Facebook o Instagram, sono considerate live streaming.

La visualizzazione di uno stream in live può essere aperta al pubblico oppure limitato ad utenti autorizzati o paganti. L’utente che fruisce del video in live streaming non ha necessità

di nessuna apparecchiatura hardware o software dedicata, basta un computer, una smart Tv, un telefono o un tablet e una buona ed affidabile connessione ad internet.

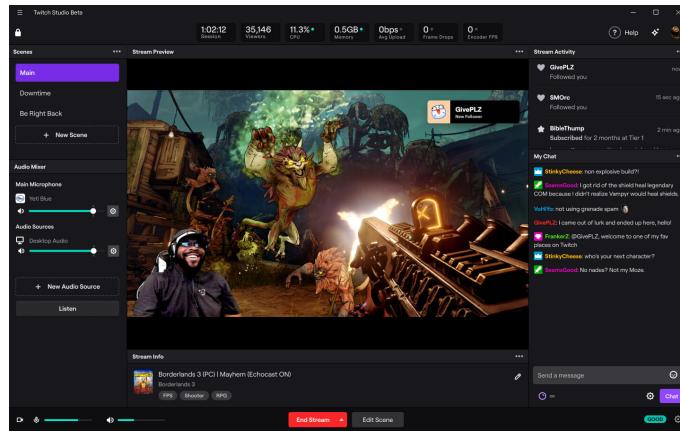


Figura 2.22: Streamer in live sulla piattaforma Twitch

La distribuzione di video in live streaming è complicata dalla varietà di piattaforme, reti di accesso e formati di streaming in concorrenza tra di loro. Gli utenti, avendo degli ottimi strumenti di connessione alla rete, si aspettano performance di alta qualità, costanti e disponibili in qualsiasi momento e con qualsiasi dispositivo, a loro scelta. Assicurare una esperienza in live del genere è una sfida irta di difficoltà, ancora ad oggi. Infatti, se il numero di spettatori supera la capacità del server, gli utenti possono essere esclusi dall'evento in live e farsi una cattiva impressione della piattaforma. Anche se i server sono in grado di gestire la domanda, potrebbero comunque esserci problemi di congestione della rete, latenza e perdita di pacchetti, costringendo il pubblico a cercare altrove una qualità del servizio migliore.

Citiamo alcune piattaforme molto utilizzate nell'ambito del live streaming di contenuti:

- **Twitch:** è la piattaforma di streaming video di Amazon, tra le più popolari del momento. La piattaforma permette di poter seguire in live streaming i creator di interesse, ma anche di poter creare dei propri live streaming. L'iscrizione alla piattaforma è gratuita e, nel piano base, consente di: scegliere chi seguire per ricevere tutti gli aggiornamenti del canale, supportare gli streamer preferiti con donazioni o tramite un abbonamento mensile al canale, utilizzare la chat per comunicare con gli altri utenti e il/i creator in tempo reale e permette anche di poter stremmare. È presente anche un piano di abbonamento premium per ricevere ulteriori vantaggi.
- **YouTube live:** è uno strumento che consente ai creator di raggiungere facilmente la propria community. YouTube live mette a disposizione vari strumenti con cui poter gestire i live streaming e interagire col pubblico. Anche in questo caso sono presenti

meccanismi di chat in diretta per consentire al pubblico di interagire col creator e con gli altri utenti.

- **Dacast:** è una piattaforma nota sia per streaming live che su richiesta, che utilizza un l'approccio SaaS. È molto utilizzata da professionisti del mondo degli affari e offre un sistema di supporto 24/7. Questo strumento è comunemente utilizzato per riunioni di grandi dimensioni o conferenze.
- **Livestream:** è una piattaforma di live streaming in grado di accettare input da molte tipologie di dispositivi. Offre una grande varietà di lettori multimediali, strumenti video e funzionalità di condivisione. Anche in questo caso c'è la possibilità di inserire una chat per aumentare la comunicazione.
- **Brightcove:** è una piattaforma basata sull'approccio SaaS, e dedicata principalmente ad una utenza professionale, essendo il costo mensile molto alto. Viene principalmente utilizzata per fini di Marketing aziendale.

2.2.5 Quali servizi di streaming scegliere

Nonostante qualche contenuto sia fruibile gratuitamente, registrandosi semplicemente alla piattaforma, la maggior parte dei servizi di streaming ha un costo.

Il primo fattore da considerare è quindi il prezzo del servizio, perché il budget personale non è un fattore da trascurare. Al di fuori del prezzo bisogna tenere conto di altri fattori:

1. Gusti personali;
2. Abitudini
3. Catalogo dei titoli disponibili;

Bisogna, infatti, considerare che ogni piattaforma ha un catalogo di titoli completamente differente da un'altra piattaforma, e quindi capire cosa ci piace di più guardare (film? serie TV? documentari?) e trovare, conseguentemente, la piattaforma che offre più contenuti relativi ai gusti personali.

Bisogna tenere conto di quanto si usufruisce del servizio, per decidere se scegliere servizi di streaming on demand in abbonamento o a consumo.

Bisogna tenere conto del/dei dispositivi su cui si fruisce del contenuto. Se, ad esempio, si guardano film sul tablet, allora le relative app devono essere ben progettate e performanti.

Infine, bisogna considerare l'ampiezza del catalogo offerto dalle piattaforme di streaming e il relativo aggiornamento, per evitare di trovarsi nelle condizioni di aver fruito di tutti i contenuti della piattaforma.

2.2.6 Lo streaming pirata

Purtroppo, esistono portali pirata che permettono la visione gratuita di contenuti a pagamento, in streaming on demand e live. Per i live streaming pirata si può effettuare una divisione dei portali in base a delle caratteristiche. Ad esempio, alcuni siti sono incentrati solamente sulla trasmissione di eventi sportivi, altri si occupano della visione di canali televisivi o della Pay-tv (come Sky). I primi possono essere organizzati per eventi da trasmettere, mettendo a disposizione del visitatore vari canali in cui poterli visionare e incorporando delle live chat in cui i propri utenti possono interagire commentando l'evoluzione dell'evento in live, creando così una vera e propria community fedele al sito. Uno dei primi e più importanti sistemi di file sharing fu Napster [8], utilizzato inizialmente per lo scambio di brani musicali in MP3. La storia di Napster fu ricca di successo ma anche di polemiche, provenienti dagli artisti che vedevano i loro guadagni ridotti e violati i propri diritti d'autore. Il sistema di Napster si basava su un protocollo peer-to-peer ibrido, con un server centrale che manteneva la lista dei sistemi connessi e dei file condivisi e gli scambi che avvenivano privatamente tra gli utenti. Nel 2001 una sentenza, solo parzialmente eseguita, obbligò Napster a pagare 26 milioni di dollari per utilizzo non autorizzato di brani e 10 milioni per diritti futuri. L'anno successivo il sito fu ufficialmente chiuso. Ad oggi è presente una nuova versione di Napster, legale e a pagamento, che si occupa dello streaming di brani musicali.



Figura 2.23: Software Napster odierno

Ad oggi i sistemi peer-to-peer sono ancora presenti ma poco utilizzati dagli utenti, sempre più spinti all'utilizzo dei servizi di streaming. Il nascere di siti e portali dedicati allo streaming

illegale ha portato ad una serie di nuove dispute legali molto complesse. Se fino a qualche tempo fa bastava commissionare multe salate per far chiudere dei server e mandare in crisi interi sistemi, oggi siti interi riescono ad aggirare la chiusura, spostandosi con rapidità su altri domini.

2.3 Concetti di intelligenza artificiale, Machine Learning e Deep Learning

2.3.1 Intelligenza artificiale (AI)

In passato gli studiosi hanno indagato diverse versioni di AI. Alcuni hanno definito l'intelligenza in termini di fedeltà alla prestazione umana, mentre altri preferiscono una definizione formale di intelligenza come razionalità, o per dirla in parole povere, "fare la cosa giusta". D'altro canto, alcuni considerano l'intelligenza una proprietà dei processi di pensiero e del ragionamento, mentre altri si concentrano sul comportamento intelligente, con una caratterizzazione esterna. Dalle due dimensioni di umano versus razionale e pensiero versus comportamento si ottengono quattro possibili combinazioni che ci aiutano a descrivere il concetto di intelligenza artificiale. Presentiamo i quattro approcci nel dettaglio [9]:

1. **Pensare umanamente:** equivale al tentativo di far sì che i computer arrivano a pensare.

Per fare ciò bisogna determinare, come prima cosa, come l'essere umano pensa. Bisogna quindi capire i meccanismi interni al cervello umano. Questo può essere svolto tramite tre metodologie:

- **Introspezione:** un atto che consiste nell'osservazione diretta e nell'analisi dell'interiorità umana, rappresentata da pensieri, pulsioni, desideri e stimoli.
- **Sperimentazione psicologica:** consiste nell'osservazione dei pensieri, pulsioni, desideri e stimoli di una persona in azione.
- **Imaging cerebrale:** consiste nell'osservazione del cervello in azione, così da poter intuire i meccanismi nervosi interni e trarne conclusioni.

2. **Pensare razionalmente:** rappresenta lo studio delle facoltà mentali attraverso l'uso di modelli computazionali. I sillogismi Aristotelici hanno rappresentato il primo tentativo di codificare formalmente il pensiero corretto. Questi sillogismi forniscono schemi di deduzione che portano sempre a conclusioni corrette, qualora siano corrette le premesse. L'obiettivo è quello di costruire un modello del pensiero razionale, basato sulle

teorie della probabilità. Queste teorie consentono di poter effettuare un ragionamento rigoroso in presenza di informazioni incerte. Tutto questo perché la logica richiede una conoscenza certa del mondo, condizione raramente vera.

3. **Agire umanamente:** la capacità di creare macchine che eseguono attività che richiedono intelligenza artificiale quando vengono svolte. Esistono diverse limitazioni matematiche riguardo la capacità di una macchina di agire umanamente. Il più conosciuto è il teorema di Gödel: esso dimostra che in qualsiasi sistema logico sufficientemente potente possono essere formulate delle proposizioni di cui non si riesce a dare una dimostrazione né di esse e ne della loro negazione, derivandone la possibilità dell'incoerenza dello stesso sistema logico. Anche il matematico Alan Turing ottenne risultati simili nel suo test per determinare se una macchina è in grado di pensare, che prese spunto dal gioco "The Imitation Game".
4. **Agire razionalmente:** un agente artificiale deve essere capace di operare autonomamente, percepire l'ambiente che lo circonda e raggiungere i propri obiettivi. Un agente razionale agisce in modo da ottenere il miglior risultato atteso, anche in situazioni in cui non si può dimostrare l'esistenza di un'azione giusta.

2.3.2 Machine Learning, Deep Learning e Reti Neurali

Con le numerose tecnologie che si utilizzano, vengono generati molti dati ogni giorno. L'insieme di questi dati viene archiviato in database digitali e rappresenta una fonte di informazione considerevole: i *Big Data*. Tuttavia, questa massa di dati costituirebbe solo un insieme di byte problematici da raccogliere e gestire, senza un trattamento adeguato. È a questo proposito che interviene il Machine Learning [10], una tecnica che permette di poter utilizzare, al meglio, questa grande mole di dati. Il Machine Learning è un sottoinsieme dell'intelligenza artificiale, che ha il compito di creare sistemi che apprendono o migliorano le performance in base ai dati che utilizzano. Il Machine Learning nasce dalla teoria che i computer possono imparare ad eseguire compiti senza essere programmati per farlo, ma seguendo degli schemi tra i dati di cui dispongono.

I tipi principali di algoritmi di Machine Learning attualmente utilizzati sono due:

1. **Machine Learning supervisionato:** con questo metodo, un data scientist agisce da guida, insegnando all'algoritmo i risultati da generare. In questo caso, l'algoritmo apprende da un set di dati già etichettato e con un output predefinito.

2. **Machine Learning non supervisionato:** utilizza un approccio più indipendente, in cui un computer impara a identificare processi e schemi complessi senza l'aiuto di un data scientist. In questo caso, l'algoritmo utilizza dati privi di etichette e per i quali non è stato definito un output specifico.

I problemi di Machine Learning possono essere divisi in base all'output che si vuole ottenere.

I più importanti sono:

- **Regressione:** rappresenta problemi in cui si deve predire il valore di una variabile continua, sulla base di valori reali, detti target. I problemi di regressione sono problemi di apprendimento supervisionato. Per la risoluzione di tali problemi vengono utilizzati algoritmi, chiamati regressori, fondati su funzioni matematiche, che aiutano il modello nella predizione dell'istanza in input.
- **Classificazione:** rappresenta problemi in cui si deve predire il valore di una variabile categorica, tramite l'utilizzo di un training set, un insieme di osservazioni per cui la variabile target è nota. Per la risoluzione di tali problemi vengono utilizzati algoritmi di classificazione che hanno lo scopo di classificare i nuovi elementi che gli vengono dati in input, sulla base del training set.
- **Clustering:** rappresenta problemi in cui si devono raggruppare elementi, che abbiano un certo grado di omogeneità ma che abbiano anche un certo grado di eterogeneità rispetto ad altri gruppi. I problemi di clustering fanno parte dei problemi di apprendimento non supervisionato, infatti, non si conosce la classe di appartenenza di un oggetto a priori.

Il Deep Learning [11] è un approccio della AI che consente ai sistemi di apprendere in base ai dati che hanno a disposizione e in base alle esperienze che il modello affronta. Il Deep Learning si basa sul modello del sistema nervoso umano, più specificamente si basa su reti neurali artificiali. Raccogliendo le diverse interpretazioni di alcuni tra i più noti scienziati impegnati nel campo del Deep Learning, potremmo definire Deep Learning un sistema che sfrutta una classe di algoritmi di apprendimento automatico che:

- Usano vari livelli a cascata per svolgere compiti di estrazione di caratteristiche e trasformazione. Ogni livello prende degli input e restituisce degli output.
- Si basano su un tipo di apprendimento non supervisionato basato su livelli gerarchici. Le caratteristiche di alto livello vengono derivate da quelle di basso livello.

- Apprendono multipli livelli di rappresentazione che corrispondono a differenti livelli di astrazione.
- Fanno parte della più ampia classe degli algoritmi di apprendimento all'interno del Machine Learning.

Gli algoritmi di Deep Learning si basano sulle reti neurali. Il loro nome e la loro struttura sono ispirati al cervello umano, imitando il modo in cui i neuroni biologici si inviano segnali, per giungere alla conclusione di un problema. Per quanto riguarda la struttura delle reti neurali, esse sono composte da livelli. È presente un livello di input, uno o più livelli nascosti e un livello di output. Ogni nodo, o neurone artificiale, si connette ad un altro e ha un peso e una soglia associati. Se l'output di qualsiasi nodo è al di sopra del valore di soglia specificato, questo viene attivato, inviando i dati al successivo livello della rete. In caso contrario non viene passato alcun dato al livello successivo della rete.

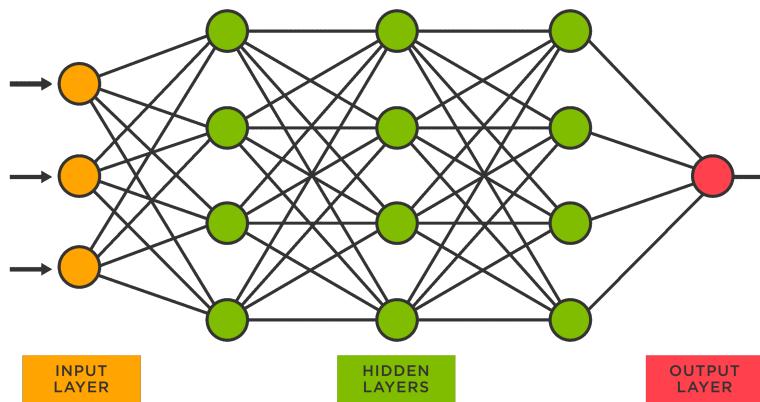


Figura 2.24: Schema di una semplice rete neurale

Le reti neurali utilizzano dati di addestramento per imparare e migliorare la loro accuratezza, nel tempo. Sono uno strumento ad alta velocità per la classificazione e organizzazione in cluster di dati. I tipi più comuni di reti neurali sono i seguenti:

- **Percettrone:** è la forma più antica di rete neurale. È la forma più semplice di rete neurale poiché contiene un unico neurone.
- **Reti neurali feed forward (o percetroni multilivello):** sono formate da un livello di input, uno o più livelli nascosti e un livello di output.

Ce ne sono di diverse tipologie:

- **Reti ricorrenti:** introducono un meccanismo di memoria grazie al fatto che i nodi sono collegati ciclicamente. I valori di uscita di uno strato, di un livello superiore,

vengono utilizzati in input a strati di livello inferiore (maggiormente utilizzato nell’ambito del riconoscimento vocale).

- **Reti neurali convoluzionali:** rappresentano un sottoinsieme di reti multistrato, formate da almeno cinque strati. È perfetta per il riconoscimento di immagini poiché si basa su una operazione matematica, chiamata convoluzione. Questa prende in input una immagine e ci applica un filtro per produrre una nuova immagine tridimensionale, di grandezza variabile a seconda delle operazioni e dei filtri.

2.4 Algoritmi di AI applicati agli streaming video

In questo capitolo discutiamo di alcuni lavori di analisi, effettuati sui flussi di video in streaming e che implementano algoritmi di intelligenza artificiale. Nello specifico discuteremo di un modello impegnato nella classificazione dei flussi di streaming video [12], un modello che si occupa di definire la qualità di esperienza nella visione di un flusso di streaming video [13] e un modello che si occupa dell’estrapolazione dei momenti salienti di una live streaming, su Twitch [14].

2.4.1 Streaming video classification using Machine Learning

L’obiettivo della ricerca è quello di creare una pipeline end-to-end per addestrare e classificare un sistema di Machine Learning che prende in input una collezione di pacchetti raccolti sull’interfaccia di rete ed è in grado di classificare i pacchetti come appartenenti a uno dei cinque servizi di streaming: YouTube, YouTube TV, Netflix, Amazon Prime, HBO. Per fare ciò verrà utilizzato il processo decisionale di Markov applicato a una rete neurale a percettore multi-nastro. La prima fase che è stata affrontata è quella della raccolta dei pacchetti in entrata e uscita dalla rete. Per fare ciò è stato utilizzato il software Wireshark prevalentemente utilizzato per la raccolta di pacchetti grezzi (PCAPS).

La seconda fase è stata quella di elaborazione dei dati; i pacchetti raccolti, infatti, sono stati analizzati e sono state estratte le informazioni importanti per la risoluzione del problema in questione. I pacchetti sono stati ordinati e ripuliti, tramite il software Pandas, che ha provveduto all’eliminazione di alcuni dati inutili all’interno dei pacchetti. Sono stati eliminati flag di sincronizzazione, di riconoscimento, di fine messaggio e altro ancora. Per una migliore gestione, i pacchetti sono stati trasformati da bit in byte. Alla fine del processo si sono ottenute 23 colonne di dati, per ogni pacchetto.

La terza fase prevede l'inserimento dei dati all'interno di una semplice rete neurale a percettore multistrato che ha il compito di creare una serie di probabilità di classificazione. L'architettura iniziale della rete neurale prevedeva: 23 nodi in ingresso, uno strato nascosto contenente 4 nodi, una funzione di rettificazione ReLU, uno strato finale di uscita a 5 nodi.

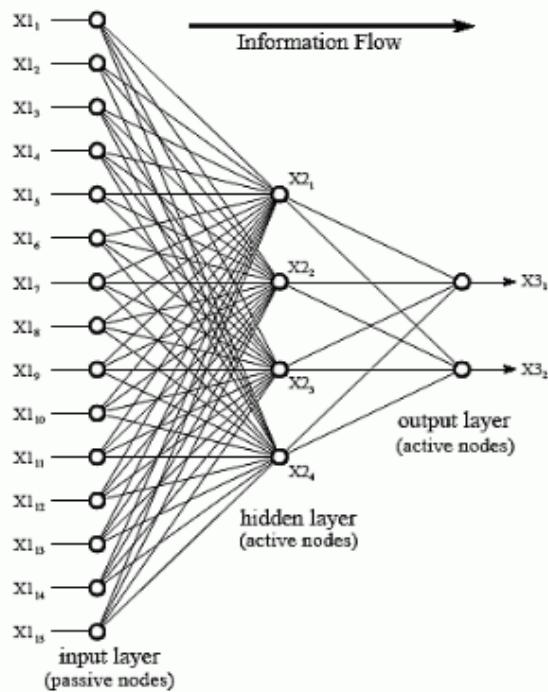


Figura 2.25: Struttura della rete neurale di partenza

La quarta fase prevede il caricamento dei dati in un processo decisionale di Markov: i migliori output della rete neurale sono stati inseriti in input ad un processo decisionale di Markov, allo scopo di migliorare l'accuratezza dei risultati della rete neurale. Tramite la libreria di MATLAB, *MatConvNet*, è stato creato un nuovo insieme di caratteristiche per ogni stato del processo decisionale di Markov. Per ogni input al processo di Markov si può passare per cinque stati. Ogni stato deve soddisfare una soglia minima di probabilità.

- Se uno stato soddisfa la soglia, si passa allo stato di selezione di tale probabilità.
- Se più stati soddisfano la condizione, verrà selezionato il massimo.
- Se nessuno stato soddisfa la soglia, si passa alla valutazione dello stato successivo.

Per trovare le soglie ottimali, è stata associata una funzione di ricompensa (aspetto caratteristico dei modelli MDP) a ciascuno degli stati di uscita, +100 per uno stato corretto e -100 per uno stato errato. È stata poi creata una politica di test iniziale in cui ogni stato aveva una

soglia minima, che la probabilità massima doveva superare. Se questa superava questa soglia la politica avrebbe classificato il pacchetto come appartenente ad un determinato insieme, altrimenti la politica passava allo stato successivo.

La quinta fase è quella relativa alla valutazione del sistema. È stato prelevato un insieme di dati per testare il sistema. Il 70% dei dati è stato usato come insieme di dati di training. Il 20% dei dati è stato utilizzato dati di validation, mentre il rimanente 10% è stato usato come insieme di dati di test. Per la valutazione dell'accuratezza del sistema è stata usata una misura in percentuale, correlata al numero di classificazioni corrette effettuate. Per quanto riguarda la rete neurale ha una accuratezza del 99.7% per i dati di training e del 73.1% per i dati di validation. È stato inserito un processo di convalida incrociata, per risolvere un problema di overfitting dovuto alla scelta dei dati e individuato dall'eccessiva variazione dell'accuratezza dei dati della rete neurale. Il processo utilizzato è il k-fold. I dati sono stati suddivisi in cinque set di convalida separati, iterando i set di convalida, in cui l'i-esimo set è stato utilizzato come test set, mentre il resto è stato utilizzato come training set. Grazie a questo processo è stato riscontrato un tasso medio di accuratezza del modello dell'81,7% (tabella 2.1).

Stream	Accuracy
Amazon Prime	0.84
Netflix	0.82
HBO	0.86
You Tube	0.77
You Tube TV	0.75

Tabella 2.1: Accuratezza della previsione per ogni servizio

Il processo decisionale di Markov ha aiutato il modello a raggiungere una accuratezza media superiore al 90%. Di seguito riportata la tabella 2.2 di visualizzazione dei risultati di accuratezza del modello.

Stream	Threshold	Accuracy
Amazon Prime	0.51	0.91
Netflix	0.75	0.92
HBO	0.6	0.902
You Tube	0.85	0.85
You Tube TV	0.9	0.845

Tabella 2.2: Tabella di accuratezza finale

2.4.2 A Deep Learning Model for Extracting Live Streaming Video Highlights using Audience Messages

L’obiettivo della ricerca è quello di creare un modello che riesca ad estrapolare automaticamente i punti salienti di un video in live streaming. In questo lavoro viene proposto un modello di Deep Learning che esamina i messaggi inviati dal pubblico estrapolando i segmenti associati ai messaggi che rilevano particolare interesse da parte del pubblico. La prima difficoltà che si affronta nell’utilizzo dei messaggi equivale alla latenza tra l’esecuzione del momento saliente e la visualizzazione della messaggistica dell’utente. Per la risoluzione di tale problema verrà utilizzata una rete neurale profonda, *Gated Recurrent Unit Deep Neural Network (biGRU-DNN)*, che si occuperà della divisione del video in una sequenza di segmenti video, esaminandone i messaggi, per etichettarli come momenti salienti. Viene usata questa tipologia di rete neurale perché:

- È efficace nell’elaborazione di dati sequenziali.
- Possiede un meccanismo di memoria che aiuta a conservare il contesto dei messaggi sequenziali dell’utente, risolvendo il problema della latenza.

Un’altra difficoltà sta nell’utilizzo dei messaggi del pubblico poiché spesso contengono slang di internet, come ad esempio le emoticon. Per risolvere questo problema è stata adottata la tecnica del word embedding, una tecnica che permette di poter rappresentare parole che hanno lo stesso significato in maniera simile, e tramite vettori di embedding, che in questo caso sono stati addestrati con una enorme quantità di messaggi internet. Il modello di estrazione degli highlight passa in una fase di apprendimento, in cui vengono raccolti un insieme di video utilizzati per l’addestramento, con i relativi messaggi del pubblico e alle sezioni di highlight etichettate. Ogni video viene diviso in una serie di segmenti e i messaggi in una sequenza di embeddings, che verranno utilizzati per addestrare il modello GRU. Si

passa poi alla fase di estrazione degli highlight in cui vengono applicate le procedure di segmentazione ed embedding ad un video di prova. Gli embeddings delle frasi relative ai segmenti del video verranno poi inserite nel modello di apprendimento per prevedere i punteggi di confidenza dei segmenti. Verranno poi selezionati i segmenti con il massimo punteggio di confidenza.

Per quanto riguarda la fase di apprendimento, ogni sezione di highlight viene divisa in una serie di segmenti sovrapposti, utilizzando un meccanismo di finestre scorrevoli. Per ogni secondo di video viene costruito un segmento video lungo quanto la lunghezza in secondi del segmento, a cui poi viene assegnato un punteggio di confidenza pari ad 1 e viene salvato come istanza di formazione positiva. Per considerare la latenza tra il momento di highlight e i messaggi del pubblico viene utilizzata una finestra di contesto, grazie alla quale poter includere le frasi poste prima e dopo il momento. Per le istanze di training negative, vengono considerati i messaggi postati prima e dopo le sezioni di highlight, più specificamente viene applicata la stessa procedura di segmentazione alle sezioni, 120 secondi prima e dopo, generando segmenti con punteggio di confidenza pari a 0.

Analizziamo ora il modello *biGRU-DNN*. Il modello è formato da due tipi di rete: una rete GRU bidirezionale a due strati e una rete DNN (*Deep Neural Network*) a più strati. La rete GRU è formata da due strati:

1. Il primo strato esamina gli embedding di frase che si verificano ogni secondo.
2. Il secondo strato elabora in maniera sequenziale gli embedding del primo strato, per produrre l'embedding del segmento.

L'embedding poi passa attraverso la rete neurale profonda (DNN) a più strati, che tramite una *funzione sigmoide* (funzione matematica limitata tra 0 e 1), prevede un valore compreso tra (0,1) che rappresenta il punteggio di confidenza del segmento di ingresso.

Per quanto riguarda l'addestramento della rete, i parametri sono inizializzati in maniera casuale. Poi, tramite il gradiente, i parametri vengono affinati.

Nella fase di estrazione degli highlight, viene impiegata la stessa procedura di elaborazione dei video, per suddividere un video di prova in una serie di segmenti sovrapposti. Tutte le frasi di embedding dei segmenti vengono inserite in sequenza nella rete appresa, per prevedere i loro punteggi di confidenza. Se si supera una certa soglia di confidenza, si prende quel segmento e si inserisce nella sequenza highlight restituita (figura 2.26).

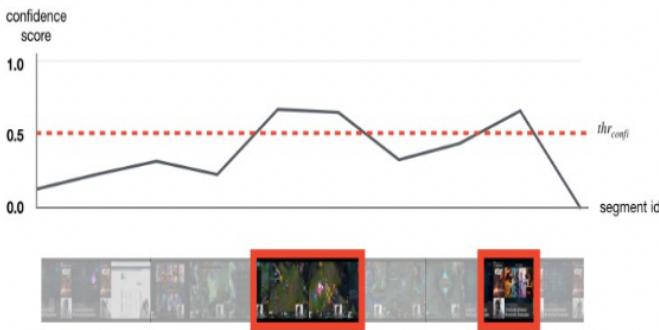


Figura 2.26: Esempio di estrazione di Highlight

Per quanto riguarda la valutazione del modello, sono stati scaricati 491 video di LOL in diretta e filtrati per durata maggiore di 40 minuti. Il 70% dei video è stato utilizzato come training set, 88 video sono stati usati come validation test per regolare i parametri del sistema, i rimanenti video sono stati utilizzati come test set.

Number of experiment videos	491
Number of training videos	353
Number of validation videos	88
Number of testing videos	50
The length of the video (sec.)	11,351,902
The length of highlight sections (sec.)	77,313
The number of user messages	37,118,352
The number of message tokens	197,169,890

Tabella 2.3: Caratteristiche dei video utilizzati per la valutazione del modello

Sono stati calcolati i parametri di *Precision* (rappresenta la precisione del modello, più specificamente è il rapporto tra il numero delle previsioni corrette di un evento e il totale delle volte che il modello lo prevede) e *Recall* (rappresenta la sensibilità del modello, più specificamente è il rapporto tra le previsioni corrette per una classe e il totale dei casi in cui si verifica effettivamente) del sistema e il modello è stato confrontato con altri modelli. Nonostante la soglia di precisione sia appena del 51.3% essa supera i metodi con cui è stata confrontata.

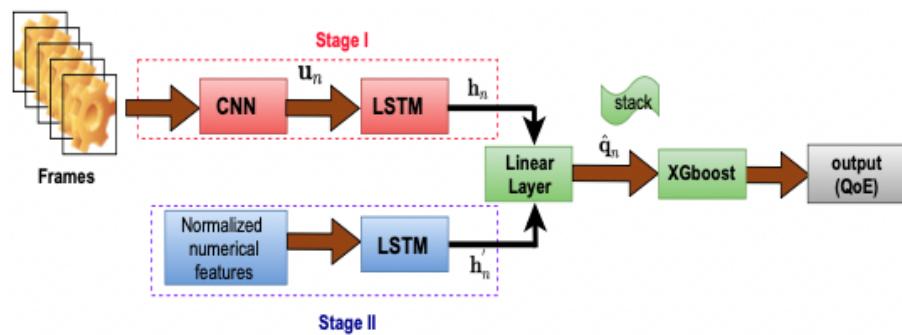
L-Char-LSTM _{100%}	0.376	0.034	0.062
L-Char-LSTM _{25%}	0.245	0.022	0.040
The message density model	0.372	0.033	0.060
biGRU-DNN	0.513	0.118	0.191

Tabella 2.4: Risultati in comparazione

2.4.3 DeSVQ: Deep Learning Based Streaming Video QoE Estimation

In questo lavoro viene proposta *DeSVQ*, un approccio Deep Learning che utilizza un framework composto da reti neurali *Convoluzionali* (CNN) e reti *Long Short Term Memory* (LSTM), ognuna delle quali cattura efficacemente le complesse dipendenze alla base del processo di previsione della qualità di esperienza di uno streaming video. I risultati di entrambe le reti vengono poi combinati linearmente e alimentati dagli alberi decisionali.

I flussi video che viaggiano sul web vengono codificati e distorti, per poi essere decodificati quando arrivano al destinatario. Nel caso di questo lavoro, i video distorti sono stati recuperati tramite un set di dati pubblici. Questi video sono serviti per calcolare le caratteristiche numeriche che verranno poi normalizzate e date in input al framework.

**Figura 2.27:** Architettura del modello

L'architettura del framework (figura 2.27) DeSVQ consiste in due stadi di ingresso. Il primo prende in input dei frame in una rete CNN, che ha il compito di estrarre le caratteristiche dai video distorti e restituire valori che andranno in input alla rete LSTM. Il secondo stadio prende in input ad una rete LSTM le caratteristiche numeriche rappresentate dalle metriche oggettive utilizzate, con lo scopo di ricavare le caratteristiche distintive che possono rappresentare le regolarità spazio-temporali dei video distorti. Avendo intervalli differenti, queste caratteristiche sono state normalizzate, prima di essere date in input alla rete. Le

caratteristiche numeriche rappresentative assumono significato grazie a una correlazione con i punteggi di QoE, come mostrato nella tabella 2.5:

Objective metrics	SROCC
PSNR	0.5247
MS-SSIM	0.6702
STRRED	-0.5495
VMAF	0.6038

Tabella 2.5: Correlazione tra le metriche oggettive e i punteggi di QoE calcolati per fotogramma

Per combinare i due stadi viene utilizzato uno strato lineare. I campioni video distorti passano attraverso uno stack di livelli convoluzionali (filtri che attraversano i fotogrammi in verticale e orizzontale ed eseguono prodotti convoluzionali). Il processo in output alla CNN va in input alla rete LSTM. La LSTM dello stadio 1 è responsabile della ricerca di relazioni tra i diversi fotogrammi video e costringe la CNN ad apprendere le caratteristiche tramite back-propagation (una tecnica di apprendimento che permette di apportare miglioramenti alla rete neurale, imparando, man mano, dagli errori che commette), mentre quella con caratteristiche numeriche (nello stadio II) trova anche le relazioni temporali tra i fotogrammi, ma in modo ristretto, coprendo solo le caratteristiche numeriche fornite in ingresso. Per far fronte all'elevata varianza dei punteggi di output target, viene utilizzato l'algoritmo *eXtreme Gradient Boosting* (XGBoost) che contiene diversi alberi decisionali. In questo algoritmo ogni albero è formato applicando il metodo di discesa del gradiente e l'ottimizzazione viene effettuata minimizzando la funzione di perdita (il nostro obiettivo) tra la QoE effettiva e quella stimata a ogni indice di fotogramma. In questo modo si ottiene la qualità video prevista per lo streaming continuo.

Per la valutazione delle prestazioni sono stati valutati diversi set di dati. Nello specifico sono stati utilizzati tre database accessibili al pubblico: *LIVE Netflix I*, *LIVE NFLX II* e *Mobile stall II*. I video dei dataset erano in formato .yuv per quanto riguarda *LIVE Netflix* e *Mobile Stall II*, in formato .mp4 per quanto riguarda *LIVE NFLX II*. La prima operazione effettuata è stata la conversione dei video da formato .mp4 in video in formato .yuv. Per quanto riguarda l'addestramento del modello, è stato suddiviso il dataset in rapporto 80:20. È stato usato un modulo esterno di Python, *PyTorch*, per rimescolare gli indici di convalida e addestramento. Il modello è stato poi convalidato con un metodo di convalida incrociata, per poter essere validato sull'intero set di dati. Per la misura di prestazione del framework, sono state utilizzate

tre misure metriche:

- Coefficiente di correlazione di Spearman Rank Order (SROCC).
- Coefficiente di correlazione lineare (LCC).
- Errore quadratico medio (RMSE).

Confrontiamo i risultati di DeSVQ con i metodi più recenti. I risultati ottenuti sul dataset LIVE Netflix I dicono che il modello raggiunge SROCC e LCC rispettivamente di 0,8935 e 0,8908. Il valore di LCC ottenuto è del 10% e del 4,8% relativamente più alto rispetto al modello di Eswara e Duc.

QoE Models	LCC	SROCC	RMSE
Bampis model	0.6741	0.5354	0.943
Eswara's model	0.8085	0.7187	0.759
Duc's model	0.8526	0.7680	0.486
Proposed DeSVQ	0.8935	0.8908	0.327

Tabella 2.6: Confronto delle prestazioni del modello DeSVQ sul dataset LIVE Netflix I con i modelli QoE esistenti

Per quanto riguarda i risultati sul dataset LIVE NFLX II, il DeSVQ batte il modello di Bampis, Eswara e Duc per quanto riguarda l'LCC rispettivamente del 20%, del 7,4% e del 6,45% e per quanto riguarda lo SROCC rispettivamente del 30%, del 9,6% e dell'8,35%.

QoE Models	LCC	SROCC	RMSE
Bampis model	0.7367	0.6783	0.789
Eswara's model	0.8276	0.8087	0.645
Duc's model	0.8355	0.8183	0.534
Proposed DeSVQ	0.8894	0.8867	0.363

Tabella 2.7: Confronto delle prestazioni del modello DeSVQ sul database LIVE NFLX II con i modelli QoE esistenti

Il modello raggiunge valori di LCC, SROCC e RMSE pari a 0,8988, 0,8936 e 0,352, rispettivamente per quanto riguarda i risultati sul dataset Mobile Stall II. Anche in questo caso, si nota un guadagno relativo dello 0,7% e dello 0,81% in termini di LCC e SROCC rispetto al modello di Duc.

QoE Models	LCC	SROCC	RMSE
Bampis model	0.7668	0.7443	0.907
Eswara's model	0.8783	0.8607	0.682
Duc's model	0.8927	0.8864	0.427
Proposed DeSVQ	0.8988	0.8936	0.352

Tabella 2.8: Confronto delle prestazioni del modello DeSVQ su Mobile Stall II con i modelli QoE esistenti

CAPITOLO 3

Applicazione sviluppata

3.1 Obiettivi e descrizione dell'applicazione

Uno degli obiettivi del lavoro di tesi è quello di creare una applicazione che permetta di poter visionare e gestire flussi di video in streaming on demand, in un contesto puramente virtuale. Infatti, l'applicazione è inquadrata nel dominio applicativo riferito all'intrattenimento, ad oggi dominio maggiormente in sviluppo in ambito di realtà virtuale. La scelta di un ambiente virtuale ha come obiettivo quello di permette all'utente di poter vivere una esperienza completamente immersiva e staccarsi completamente dalla realtà circostante, entrando in una nuova realtà, in cui potersi rilassare. Infatti, l'utente viene generato al centro di una scena che rappresenta un tipico ambiente rilassante (figura 3.1): un salotto di casa, luci moderate e un proiettore posto di fronte ad esso. A sinistra dell'utente è presente un sofà e un tavolino da soggiorno, contenente svariati oggetti con cui poter interagire. A destra dell'utente è presente un tavolo contenente vari oggetti, compreso il telecomando grazie al quale poter interagire col proiettore. L'applicazione permette all'utente di muoversi liberamente all'interno della stanza, che presenta anche uno spazio contenente una piccola cucina, come si vede in figura 3.2



Figura 3.1: Stanza in cui viene generato l'utente



Figura 3.2: Stanza rappresentante la cucina, facilmente raggiungibile dall'utente

Il proiettore ha come obiettivo quello di fornire un video player, esteticamente bello e minimale, utile alla riproduzione e gestione dei contenuti video. Il proiettore consente di poter effettuare le seguenti operazioni sul flusso in streaming:

- **Stop:** permette di bloccare e terminare l'esecuzione del contenuto multimediale.
- **Play/Pause:** permette di impostare momentaneamente il contenuto in pausa e riprenderne in seguito la visione.
- **Next:** permette di inviare una richiesta e visionare il contenuto multimediale seguente.
- **Previous:** permette di inviare una richiesta e visionare il contenuto multimediale precedente.
- **Next 15 seconds:** permette di avanzare la riproduzione del video attualmente in esecuzione di 15 secondi.
- **Previous 15 seconds:** permette di retrocedere la riproduzione del video attualmente in esecuzione di 15 secondi.

Parte integrante del proiettore è la barra di progresso del video, di colore rosso, che ha come obiettivo quello di indicare all'utente lo stato di avanzamento relativo al video in riproduzione, come si vede in figura 3.3.

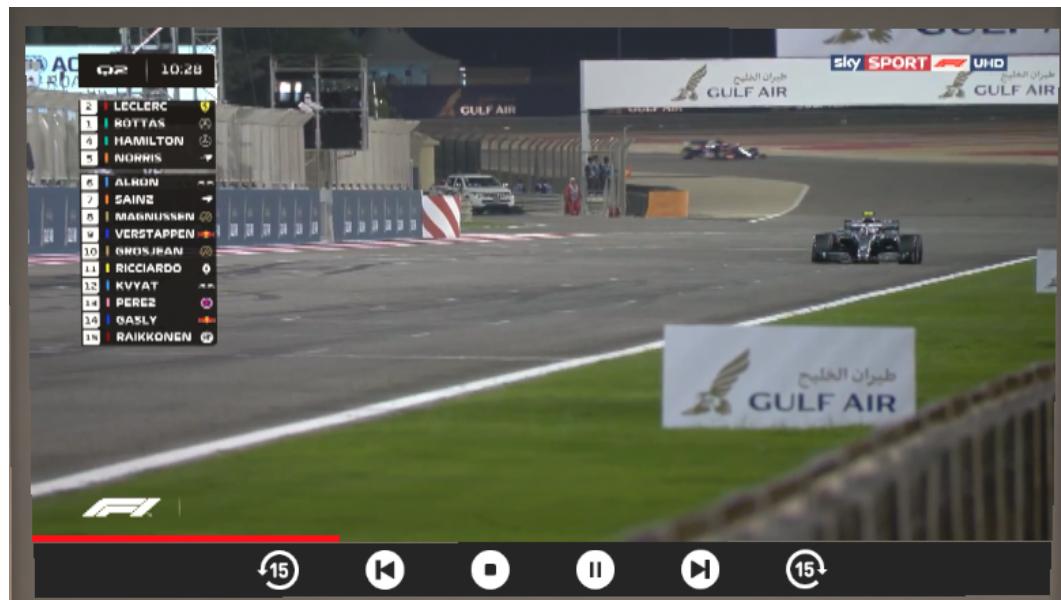


Figura 3.3: Progress bar e barra di interazione del video player

L'interazione tra utente e proiettore avviene tramite un oggetto che funge da telecomando virtuale. Quando l'utente prende il controllo di questo oggetto, viene attivato un raggio di colore rosso, nella parte anteriore e centrale dell'oggetto, che si estende all'interno della stanza e permette all'utente di comprendere il raggio di azione del telecomando, utile a semplificare l'interazione con il proiettore (vedere figura 3.4).

Quando l'utente punta elementi della barra di interazione, gli viene restituito un feedback optico hardware, grazie al controller relativo alla mano in cui è attualmente presente l'oggetto virtuale. Questi piccoli accorgimenti sono stati inseriti in modo da rendere più gradevole l'esperienza utente.



Figura 3.4: Interazione col video player, lato utente

3.2 Specifiche tecniche dell'applicazione

In questo paragrafo analizzeremo tutti gli oggetti, con le relative componenti, costituenti l'applicazione sviluppata.

Il fulcro dell'applicazione è l'utente, esso si muove all'interno della scena e interagisce con gli oggetti tramite delle mani virtuali, che l'utente gestisce tramite il controller destro e sinistro di Oculus Quest 2. L'utente è rappresentato da un oggetto prefab (prefabbricato/precostruito) che funge da avatar, ottenuto grazie all' implementazione dell'SDK (software development kit) che Oculus mette a disposizione agli sviluppatori. Questo oggetto si presenta col nome di *OVRPlayerController* e contiene tutti i settaggi necessari alla rappresentazione dell'utente e

delle due mani all'interno della scena virtuale. L'oggetto conteneva però dei piccoli problemi per quanto riguarda gli script che permettessero il corretto impossessamento degli oggetti. Gli oggetti, infatti, non avevano una ben precisa direzione di presa, e questo creava problemi quando ci si impossessava dell'oggetto telecomando e si voleva interagire con il video player. Lo script *OVRGrabber*, infatti, è stato sottoposto ad una analisi che ha portato conseguentemente a delle piccole modifiche, in modo da ottenere una adeguata direzione di impossessamento degli oggetti.

Nella scena sono presenti molteplici oggetti, con i quali l'utente può interagire. Agli oggetti con cui viene permessa l'interazione, è stato inserito un opportuno script, laddove necessario, utile a definire il comportamento di tali oggetti, e sono state inserite le opportune componenti utili a definirne la fisica, nello specifico sono state utili le componenti:

- Rigidbody, per dotare gli oggetti di gravità.
- Collider, per impostare le dimensioni fisiche dell'oggetto.

Per poter visionare i flussi video provenienti da internet, all'interno della scena è stato inserito un oggetto, chiamato proiettore, che ha il compito di effettuare una richiesta HTTP ad un bucket online AWS. Il Bucket restituisce un array di stringhe JSON contenenti i metadati relativi ad ogni video. È presente poi, come oggetto figlio del proiettore, un video player a cui viene dato in input l'URL riferito al video che deve essere visionato e che si occupa di riprodurre il contenuto a schermo. Il bucket utilizzato viene fornito tramite il servizio di storage ,in Cloud, di Amazon: *Amazon S3*. Ogni video, all'interno del bucket, contiene i seguenti metadati:

- **name:** nome del video.
- **urlS3:** link al video generico, in formato Raw.
- **urlCDN:** link al video localizzato.
- **category:** categoria di appartenenza del video.

L'oggetto telecomando è stato creato con Blender, un software di modellazione 3D, tramite il quale è stato possibile creare la forma dell'oggetto tridimensionale e impostare un adeguato materiale di cui era composto tale oggetto, modificandone le caratteristiche nei minimi dettagli. Anche l'oggetto telecomando ha associato uno script che gli permette di eseguire determinate operazioni. Quando l'utente prende possesso del telecomando, viene lanciato

un raggio, grazie alla componente Line Renderer, che permette all'utente di capire dove si sta puntando.

Listing 3.1: Script Telecomando.cs

```
1 RaycastHit hit;
2 if (Physics.Raycast(transform.position, transform.forward, out hit, distance,
3     mask))
4 {
5     cast = hit.collider.gameObject;
6     ComandoPlayer comandoPlayer = hit.collider.GetComponent<ComandoPlayer>();
7     if(comando != comandoPlayer)
8     {
9         if (hapticFeedback == null)
10        {
11            hapticFeedback = HapticFeedback(grabController);
12            StartCoroutine(hapticFeedback);
13        }
14        comando = comandoPlayer;
15        raggio.SetPosition(1, hit.point);
16    }
17 else
18 {
19     comando = null;
20 }
21 if(OVRInput.GetDown(inputId, grabController))
22 {
23     if (comando)
24     {
25         comando.Execute();
26     }
27 }
```

Il telecomando può interagire solamente con gli oggetti contenuti nella barra di interazione del proiettore. Per realizzare ciò, è stato creato un layer dedicato a questi oggetti ed è stata poi settata una layermask, ossia un vincolo di tracciamento relativo al raggio lanciato dal telecomando, che ha come obiettivo quello di rendere visibili, al raggio, solamente gli oggetti contenuti all'interno della barra di interazione. Infatti, come si può notare nello snippet di codice 3.1, viene utilizzato un RaycastHit che ha il compito di lanciare una raggio, dalla posizione dell'utente in avanti, ad una determinata distanza massima e rilevando i collisori

presenti all'interno della maschera che si sta utilizzando, quella prima descritta. Viene poi salvato il GameObject con cui si collide (riga 2-5). Alla riga 5 viene, poi, salvato il comando con cui si sta collidendo (play, pause, next...), e viene effettuato un controllo su tale, per capire se ci si sta spostando sullo stesso comando. Nel caso in cui ci si stesse spostando su un comando, e non è lo stesso, viene lanciato un feedback aptico e una coroutine per controllarlo (codice 3.2). Viene così trasmesso all'utente la sensazione che l'elemento puntato è cliccabile.

Le righe 21-27 vengono chiamate in causa quando l'utente clicca un comando all'interno della barra di interazione del video player. In questo caso viene lanciato un evento che permette di eseguire il comando lanciato.

Listing 3.2: Coroutine feedback aptico

```
1 private IEnumerator hapticFeedback = null;
2 private IEnumerator HapticFeedback(OVRInput.Controller grabController)
3 {
4     OVRInput.SetControllerVibration(1, 1, grabController);
5     yield return new WaitForSeconds(0.01f);
6     OVRInput.SetControllerVibration(0, 0, grabController);
7     hapticFeedback = null;
8 }
```

La barra di interazione col video player prende input solo dall'oggetto telecomando. Infatti, come sopra descritto, in caso di click di un elemento della barra di interazione, viene lanciato un evento, descritto nel codice 3.3.

Listing 3.3: Script ComandoPlayer.cs

```
1 public class ComandoPlayer : MonoBehaviour
2 {
3     public VideoClipManager.ActionPlayer azionePlayer;
4     public UnityEvent<VideoClipManager.ActionPlayer> evento;
5
6     void Start() { }
7
8     void Update() { }
9
10    public void Execute()
11    {
12        if(evento != null)
13        {
14            evento.Invoke(azionePlayer);
15        }
16    }
17}
```

```
15         }
16     }
17
18     private void OnDestroy()
19     {
20         evento.RemoveAllListeners();
21     }
22 }
```

Questo evento passa il controllo all'oggetto video player che, tramite lo script VideoClipManager, modifica lo stato del video. Lo stato del video viene rappresentato tramite un tipo enumeratore, i seguenti stati raggiungibili:

- **Init**: stato di default iniziale, si ottiene quando il video non è stato ancora richiesto
- **Preparing**: il video è stato richiesto ed è in fase di preparazione.
- **Play**: il video è in play.
- **Pause**: il video è in pausa.
- **Stop**: il video è stoppato.

Quando viene richiesto un video, si passa dallo stato Init allo stato Preparing. Quando il video è pronto, si passa allo stato Play e viene avviata la progress bar relativa al video (codice 3.4).

Listing 3.4: Script VideoClipManager.cs, metodo Update()

```
1 void Update()
2 {
3     switch (videoState)
4     {
5         case State.Init:
6             break;
7
8         case State.Preparing:
9             if (videoPlayer.isPrepared)
10            {
11                ChangeState(State.Play);
12            }
13            break;
14     }
```

```

15     if(progressBar != null)
16     {
17         if(videoPlayer.frameCount > 0)
18         {
19             progressBar.fillAmount = (float)videoPlayer.frame / (float)
20                 videoPlayer.frameCount;
21         }
22     }

```

All'arrivo di un comando di cambio di stato, lo script `VideoClipManager` preleva il comando richiesto e chiama il metodo adibito al quel cambio di stato. I metodi adibiti a fare ciò sono i medesimi:

- **SkipVideo** (codice 3.5): viene chiamato quando l'utente chiede di retrocedere o avanzare di 15 secondi la riproduzione del video.
- **ChangeClip** (codice 3.6): viene chiamato quando l'utente richiede un nuovo video.
- **ChangeState** (codice 3.7): viene chiamato per impostare il video in play, pausa, stopparlo.

Listing 3.5: Script `VideoClipManager.cs`, metodo `SkipVideo()`

```

1 void SkipVideo(float value)
2 {
3     double newTime = videoPlayer.time + value;
4     if(newTime < 0)
5         newTime = 0f;
6     else if (newTime >= videoPlayer.length)
7     {
8         newTime = videoPlayer.length;
9     }
10    videoPlayer.time = newTime;
11 }

```

Listing 3.6: Script `VideoClipManager.cs`, metodo `ChangeClip()`

```

1 void ChangeClip(int direction)
2 {
3     indiceArrayVideo = (indiceArrayVideo + direction) % vods.video.Count;

```

```
4     if(indiceArrayVideo < 0)
5     {
6         indiceArrayVideo = vods.video.Count - 1;
7     }
8     videoPlayer.url = vods.video[indiceArrayVideo].urlCDN;
9     ChangeState(State.Preparing);
10 }
```

Listing 3.7: Script VideoClipManager.cs, metodo ChangeState()

```
1 void ChangeState(State stato)
2 {
3     switch (stato)
4     {
5         case State.Play:
6             videoPlayer.Play();
7             isStopped = false;
8             break;
9         case State.Pause:
10            videoPlayer.Pause();
11            isStopped = false;
12            break;
13         case State.Stop:
14             if(!isStopped)
15             {
16                 isStopped = true;
17             }
18             videoPlayer.Stop();
19             break;
20         case State.Preparing:
21             isStopped = false;
22             videoPlayer.Prepare();
23             break;
24     }
25     videoState = stato;
26 }
```

Alle funzioni play e pause è associato un unico comando e, per consentire un riscontro visivo relativo al cambio di stato, del contenuto in esecuzione, all'utente, è stato inserito un ulteriore script che permette di effettuare uno switch relativo alle icone di play e pause (codice 3.8).

Listing 3.8: Script PlayPauseIcon.cs

```
1 public void Execute()
2 public class PlayPauseIcon : MonoBehaviour
3 {
4     public GameObject playIcon;
5     public GameObject pauseIcon;
6
7     public void ShowPlay()
8     {
9         if(playIcon)
10            playIcon.SetActive(true);
11         if(pauseIcon)
12            pauseIcon.SetActive(false);
13     }
14
15     public void SwitchIcons ()
16     {
17         if(playIcon)
18             playIcon.SetActive(!playIcon.activeInHierarchy);
19         if(pauseIcon)
20             pauseIcon.SetActive(!pauseIcon.activeInHierarchy);
21     }
22 }
```

Le luci sono una componente importante della scena, sono state ben organizzate in modo da creare un ambiente rilassante. Ho avuto modo di inserire svariate tipologie di luci all'interno della scena [15]:

- **Directional light:** luce adatta a simulare l'illuminazione esterna, poiché i suoi raggi raggiungono distanze infinite (figura 3.5). La posizione della directional light non influenza la luce all'interno della scena. La rotazione sull'asse Y, invece, influenza l'illuminazione della stanza.

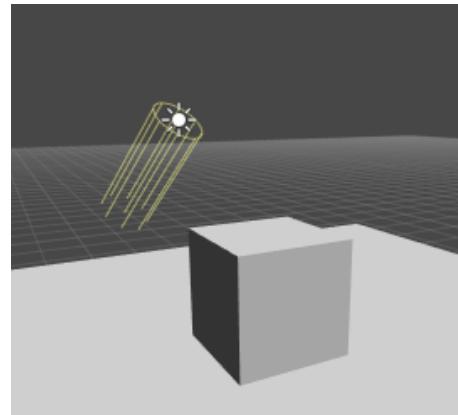


Figura 3.5: Direction light

- **Spotlight:** sono dei faretti che proiettano un cono di luce e possono raggiungere una certa ampiezza e distanza massima (figura 3.6). Per questa tipologia di luce è importante la posizione in scena e la rotazione poiché ogni minimo movimento influisce sull'illuminazione della scena.

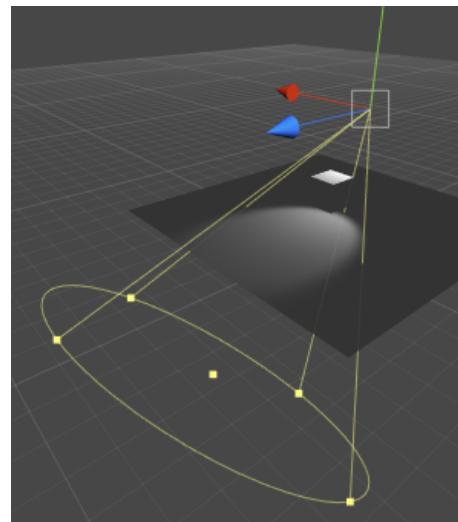


Figura 3.6: Spotlight

- **Reflection probe:** sono luci che catturano una vista sferica dell'ambiente circostante, memorizzano poi le immagini catturate come mappe cubiche utili per oggetti con materiali riflettenti (figura 3.7). Il risultato di questo è che le luci che riflettono sugli oggetti possono cambiare, in modo convincente, in base all'ambiente.

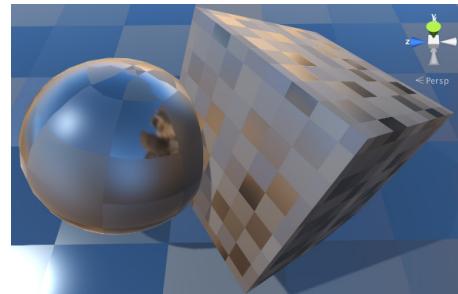


Figura 3.7: Reflection probe

Il rendering delle luci, all'interno della stanza, è stato effettuato tramite un processo chiamato Backing. Il Backing [16] è un processo, utilizzato per oggetti statici, che permette di effettuare un rendering delle luci ed ombre sulle lightmap. Per lightmap si intende una texture map che contiene i valori delle luci ed ombre proiettate sugli oggetti della scena. Trattandosi di un processo che agisce sugli oggetti statici della scena, il Backing viene effettuato in fase di creazione del gioco e non durante il gameplay. Prima di effettuare tale processo, gli oggetti della scena devono essere in qualche modo preparati.

Per quanto riguarda l'applicazione creata, su ogni oggetto coinvolto nel processo di Backing sono state effettuate queste operazioni:

- Ogni oggetto contenente una lightmap è stato reso statico.
- Ogni oggetto contenente un lightmap ha un set di coordinate UV per la lightmap.

All'interno della scena è stata inserita una Skybox [17], che si estende intorno all'intera scena e ha il compito di mostrare l'aspetto del mondo all'orizzonte. La skybox scelta (figura 3.8) rappresenta un tramonto, ottima per dare un'atmosfera rilassante alla scena, ed è stata estrapolata dall'asset *Skybox Series Free* trovato all'interno dell'Unity Asset Store.

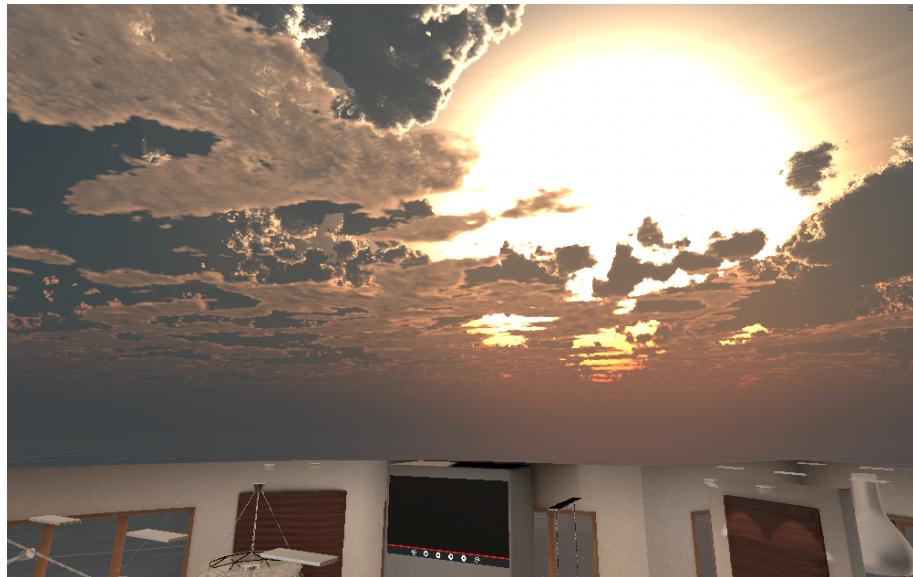


Figura 3.8: Skybox scelta per la scena

3.3 Strumenti e tecnologie utilizzate

La creazione di questa applicazione è stata affidata al software Unity [18], risorsa molto utile alla creazione di esperienze in 2D, 3D o Virtual Reality. Unity è un motore grafico, che permette la creazione di videogiochi esportabili su più piattaforme: desktop, web e diversi dispositivi (mobile e console). Infatti, molti sviluppatori indipendenti utilizzano Unity, in modo da distribuire i loro prodotti su un maggior numero di mercati. Unity fornisce un ambiente di sviluppo visuale, che permette di gestire molteplici oggetti, la cui parte logica può essere scritta in C#, JavaScript e/o Boo. Unity è disponibile in due versioni: gratuita e pro. Quella utilizzata per la creazione dell'applicazione è la versione gratuita, una versione già molto completa e che fornisce strumenti per la creazione ed esportazione di giochi multipiattaforma. La versione pro include alcuni strumenti avanzati e tool per le performance.

Gli oggetti hanno delle proprie caratteristiche fisiche e un comportamento, impostato tramite gli script del linguaggio C sharp. C sharp, come cita il libro “The c# programming Language” [19] è un linguaggio di programmazione semplice, moderno, orientato agli oggetti e type-safe. Fa parte della famiglia dei linguaggi che hanno come radice il linguaggio C, ed è quindi molto familiare ai programmati che utilizzano C, C++ o Java. Nonostante sia un linguaggio orientato agli oggetti, C# include anche il supporto per la progettazione orientata alle componenti (sempre più utilizzata nei software moderni, poiché presentano un modello di programmazione con proprietà, metodi ed eventi).

Alcune caratteristiche rendono le applicazioni costruite con linguaggio c# robuste:

- **Garbage collection:** ha lo scopo di recuperare memoria altrimenti utilizzata da oggetti non coinvolti per un certo periodo di tempo, all'interno dell'applicazione.
- **Gestione delle eccezioni:** fornisce un approccio strutturato ed estendibile per il rilevamento e la gestione degli errori.
- **Type-safe:** impossibilita la lettura di variabili non inizializzate, l'inizializzazione di array oltre il loro limite o ancora di eseguire cast non autorizzati.

Tutti i tipi di C# ereditano da un oggetto principale, ciò significa che condividono un insieme di operazioni comuni, inoltre vengono supportati sia i tipi definiti dall'utente che i tipi di valore consentendo l'allocazione dinamica degli oggetti e la memorizzazione in linea di strutture leggere.

Come IDE di scrittura degli script dell'applicazione è stato utilizzato Visual Studio Code. Utili se sue estensioni che permettono di migliorare e semplificare l'esperienza di scrittura del codice. Per questa applicazione, per un buon supporto da parte di Visual Studio Code, è stato utile installare l'estensione *c# for Visual Studio Code (powered by OmniSharp)*.

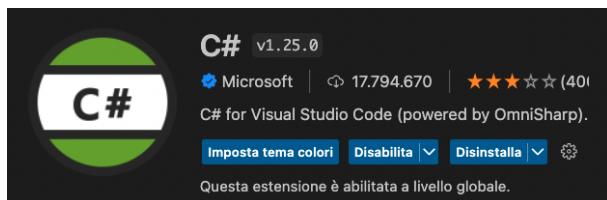


Figura 3.9: Estensione utilizzata in Visual Studio Code

L'oggetto che gestisce l'interazione col video player assume la forma di un telecomando, complicata da realizzare tramite Unity. Per realizzare una buona rappresentazione tridimensionale di tale oggetto è stato utilizzato Blender, uno dei software più utilizzati in ambito di modellazione 3D. Infatti, come annuncia il libro "Mastering Blender" [20], il numero di utenti utilizzatori del software Blender sta esponenzialmente aumentando. Il numero di libri e DVD di formazione, relativi a questo software, è passato da 0 libri e DVD nel 2006 ad un numero molto elevato di libri e DVD di formazione creati da case editrici di maggiore e minore importanza, che raccontano l'animazione, la visualizzazione architettonica, la simulazione fisica e l'uso generale del software. Anche le dimensioni del codice di base di Blender sono quasi raddoppiate dagli inizi del 2005 ad oggi, grazie al crescente interesse degli sviluppatori di computer grafica di tutto il mondo.



Figura 3.10: Rendering, in Blender, dell'oggetto telecomando

Ultimo, ma non per importanza, è stato utilizzato il software GitHub per il salvataggio degli aggiornamenti giornalieri relativi alla scrittura dell'applicazione e poter usufruirne su diverse piattaforme hardware.

3.4 Architettura dell'applicazione

Costruiamo, in questo paragrafo, uno schema che rappresenta la struttura architettonica relativa al flusso principale di interazione con l'applicazione sviluppata, ossia la richiesta e il conseguente ricevimento del flusso multimediale da riprodurre.

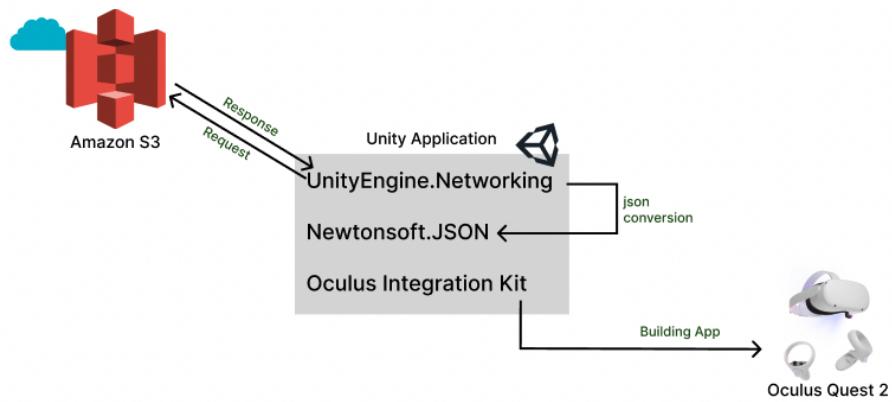


Figura 3.11: Architettura dell'applicazione sviluppata

Fondamentali nella realizzazione di questa applicazione sono stati:

- **API UnityEngine.Networking:** nello specifico è stato utile l'utilizzo della classe UnityWebRequest [21], una classe che fornisce metodi per la gestione di un flusso di

comunicazione HTTP, con un server online. Nel caso specifico dell'applicazione sviluppata, la richiesta HTTP è stata effettuata ad un bucket online AWS, contenente un array di stringhe JSON con informazioni relative ai vari video da visualizzare. La richiesta HTTP può essere effettuata con metodo get, post o put; In questo caso è stato utilizzato il metodo get. Quando la richiesta è pronta, per provvedere all'invio di tale richiesta viene utilizzato il metodo *SendWebRequest*. Per questa applicazione è stato opportuno l'inserimento della richiesta HTTP all'interno di una coroutine, per evitare lag dell'applicazione. È consigliato utilizzare la coroutine quando si itera su una collezione di dati o si accede a file di grandi dimensioni, essa infatti permette di interrompere il processo in un momento specifico, restituire la parte di dati ricevuta fino a quell'istante e ricominciare, quando utile, dallo stesso punto in cui si è interrotto il processo. Per quanto riguarda invece il download o upload di dati sono utili i metodi *DownloadHandler* e *UploadHandler*.

Listing 3.9: Coroutine per effettuare una richiesta HTTP

```
1  private IEnumerator GetText()
2  {
3      UnityWebRequest request = UnityWebRequest.Get(textURL);
4      yield return request.SendWebRequest();
5      if (request.isHttpError || request.isNetworkError)
6      {
7          Debug.LogError(request.error);
8      }
9      else
10     {
11         var text = request.downloadHandler.text;
12         vods = JsonConvert.DeserializeObject<Vods>(text);
13         if(vods != null)
14         {
15             videoPlayer.url = vods.video[0].urlCDN;
16             ChangeState(State.Preparing);
17         }
18     }
19 }
```

- **API Newtonsoft.JSON:** nello specifico è stato utile utilizzare il metodo *DeserializeObject* (seconda parte codice 3.9), grazie al quale ho potuto trasformare la stringa JSON in un

oggetto .NET, che contiene le informazioni relative ai video da visualizzare. L'oggetto in cui veniva deserializzata la stringa JSON è formato dai seguenti campi:

- **name**: rappresenta il riferimento per il nome del video.
- **urlS3**: rappresenta il riferimento per il link al video generico, in formato Raw.
- **urlCDN**: rappresenta il riferimento per il link al video localizzato.
- **category**: rappresenta il riferimento per la categoria di appartenenza del video.

Listing 3.10: Classe di appoggio per la deserializzazione della stringa JSON

```
1 [System.Serializable]
2 public partial class Vods
3 {
4     [JsonProperty("vods")]
5     public List<Vod> video;
6
7 }
8
9 [System.Serializable]
10 public partial class Vod
11 {
12     public string name;
13     public string urlS3;
14     public string urlCDN;
15     public int year;
16     public string category;
17 }
```

-
- **Oculus Integration Kit [22]**: offre un supporto per lo sviluppo di rendering, social, piattaforma, audio e avatar per i dispositivi di realtà virtuale di Oculus, e alcuni altri dispositivi VR. Il Kit di Oculus è molto sostanzioso, mostriamo di seguito alcuni elementi contenuti al suo interno:
 - **Audio Manager**: contiene gli script per la gestione degli effetti audio e sonori dell'applicazione.
 - **Avatar**: contiene gli script e i prefab per l'aggiunta di avatar all'interno dell'applicazione.
 - **LipSync**: contiene una serie di plugin e script utili per la sincronizzazione dei movimenti delle labbra degli avatar, con i suoni del parlato.

- **Sample Framework:** alcune utili implementazioni dei concetti di interazione.
- **VoiceMod:** contiene una serie di plugin per modificare i segnali audio in entrata.
- **VR:** contiene le utility di Oculus VR, un insieme di script e prefab per consentire lo sviluppo di applicazioni in VR.

Per questa applicazione, è stato utile utilizzare le componenti presenti nel pacchetto VR e *Sample Framework*. Uno dei prefab utili è l'*OVRPlayerController* che permette al giocatore di muoversi nell'ambiente virtuale, ed include componenti figlie necessarie per il controllo 3D. Include anche il prefab *OVRCameraRig*, che funge da telecamera VR. Altri prefab importanti sono stati *CustomHandLeft* e *CustomHandRight* del pacchetto *Sample Framework*.

Per quanto riguarda gli script forniti da Oculus, alcuni importanti per l'applicazione sono stati *OVRGrabber*, che permette alle mani virtuali di port interagire con oggetti all'interno della scena e *OVRGrabbable* che permette agli oggetti, presenti nella scena, di poter essere raccolti.

3.5 Valutazione preliminare dell'applicazione

L'applicazione è stata sottoposta ad una utenza che ha avuto l'opportunità di utilizzarla e raccontare la loro esperienza di utilizzo. Essendo l'applicazione sviluppata all'interno di un team di sviluppo software in realtà virtuale, si è ritenuto opportuno sottoporre, ai membri del team, la suddetta applicazione, in modo da ottenere dei riscontri da persone appartenenti al settore.

Il test dell'applicazione aveva l'obiettivo di rispondere a queste domande:

1. Come appare la scena all'utilizzatore dell'applicazione? È una scenario rilassante o sembra confusionario?
2. È soddisfacente il movimento dell'avatar all'interno della scena? Sembra camminare troppo veloce? Fa movimenti bruschi?
3. L'interazione con gli oggetti della scena sembra, fisicamente, ben progettata?
4. Come avviene l'interazione con il videoplayer? È rapida? È soddisfacente? È ben progettata?
5. La risoluzione dei contenuti multimediali è alta, mediocre o bassa?

6. L'applicazione fa il lavoro per cui è stata progettata?

La scena è stata percepita come un ambiente rilassante, è stato infatti apprezzato il rendering delle luci interne e il tramonto che, entrando in casa, creava un mix di colori molto rilassanti.

L'interazione con gli oggetti della scena non ha avuto contestazioni da parte degli utenti, risultando rapida e piacevole.

Per quanto riguarda l'interazione col video player, è stato molto apprezzato il feedback aptico restituito dal controller di Oculus quando ci si muoveva sugli elementi della barra di interazione del video player. Anche la manipolazione del video è stata apprezzata, risultando fluida e rapida, limitata all'unico vincolo relato alla velocità di banda della connessione internet, trattandosi di video in streaming on demand.

Per quanto riguarda i flussi in streaming che venivano visualizzati, secondo gli utenti riuscivano a mantenere una buona qualità video e audio, non ci sono state quindi feedback negativi a riguardo.

L'unica contestazione ricevuta, da parte di qualche utente, è relativa al movimento direzionale dell'avatar che, sembrava essere troppo veloce e poco fluido, causando la sensazione del *motion sickness* (spiegato nello stato dell'arte, paragrafo 2.1.9). Questo rappresenta il movimento di default dei player che ci viene fornito dall'SDK di Oculus Quest che non è stato molto analizzato poiché non rientrava nella parte di interesse del progetto.

CAPITOLO 4

Conclusioni

4.1 Sviluppi futuri

L'applicazione sviluppata apre ad una enorme quantità di sviluppi futuri. In seguito, verranno riportate alcune idee da aggiungere all'app esistente per renderla sempre più completa.

4.1.1 Google-Speech-To-Text

Una possibile idea potrebbe essere quella di implementare il supporto ai comandi vocali [23], in modo da trasferire comandi al proiettore direttamente con la voce, evitando di utilizzare il telecomando, un po' come succede attualmente con i più moderni telecomandi, che permettono di poter inviare un comando alla propria smart TV parlando direttamente al telecomando o alla smart TV. Per implementare i comandi vocali potrebbe essere una idea quella di utilizzare le API di Google Cloud. Nello specifico è possibile utilizzare le API Google Speech-To-Text. Questa è una API appartenente al *Conversational AI* di Google Cloud che ha il compito di convertire la voce proveniente dal microfono in un testo, utilizzabile all'interno dell'applicazione. Per poter usufruire di questa API all'interno di Unity bisogna:

- Sottoscrivere un abbonamento a Google Cloud con costo variabile in base all'utilizzo delle API che Google Cloud fornisce.

- Creare un nuovo progetto in Google Cloud e richiedere accesso alla libreria Google-Speech-To-Text.
- Inserire la libreria come package in Unity (il modo più semplice per installarlo è tramite terminale).
- Creare in Unity una cartella denominata StreamingAssets e inserire all'interno il file JSON che Google Cloud ci fornisce.
- Creare un GameObject in Unity e associarvi la componente StreamingRecognizer, che permette la scelta della periferica input da utilizzare, qualora ce ne fosse più di una.

4.1.2 DialogFlow

Un'altra possibile idea potrebbe essere quella di implementare un fattore di compagnia all'interno dell'applicazione, in modo da coinvolgere maggiormente l'utente utilizzatore. Potrebbero infatti essere inseriti degli agenti virtuali che implementano le capacità di dialogare con l'utente che sta utilizzando l'applicazione. Per ottenere una conversazione articolata potrebbero essere costruite delle interfacce conversazionali tramite le API di DialogFlow [24], facente parte dell'offerta *Conversational AI* di Google Cloud. DialogFlow è una piattaforma di comprensione del linguaggio naturale che semplifica la progettazione ed integrazione di interfacce utente conversazionali in qualsiasi tipologia di applicazione (mobile, web...). DialogFlow consente di poter analizzare diversi tipi di input dei clienti, compresi quelli audio o testuali, e rispondere in molteplici modi: in maniera testuale o naturale.

DialogFlow fornisce due tipi di agenti virtuali:

- **DialogFlow CX (customer experience):** offre la progettazione di agenti virtuali complessi, adottando un approccio basato su macchine a stato per il controllo dei percorsi delle conversazioni, mentre i flussi e le pagine sono gli elementi costitutivi della progettazione delle conversazioni. Questo offre un controllo chiaro ed esplicito sulla conversazione con l'utente. Un agente DialogFlow CX comprende il linguaggio naturale degli utenti, traduce il testo o l'audio dell'utente in dati strutturati che l'app è in grado di comprendere.
- **DialogFlow ES (Essentials):** è la versione precedente a quella CX. È il tipo di agente standard, adatto ad agenti di piccole e medie dimensioni e di complessità da semplice a moderata. I parametri intent (parametri che categorizzano gli intenti di una conversa-

zione, per l’utente) sono gli elementi costitutivi della progettazione conversazionale e i contesti sono utilizzati per controllare i percorsi della conversazione.

Il funzionamento di DialogFlow si fonda sulla creazione di bot che imparano sulla base di modelli che evolvono automaticamente. Grazie al machine learning viene compreso l’intento dell’utente che parla o scrive.

4.1.3 Estrapolazione dati in tempo reale

Un’altra possibile idea potrebbe essere quella di implementare un sistema capace di estrarre dati relativi ad un evento in streaming e inserire una sezione di visualizzazione di tali dati all’interno dell’applicazione, per migliorare l’esperienza utente. Un esempio di utilizzo potrebbe essere l’evento di Formula 1, in cui si potrebbero estrarre i dati raccolti durante la gara per ottenere informazioni aggiuntive sull’evento (condizioni delle automobili, condizione della pista, classifica, tempo migliore, ecc.).

4.1.4 Introduzione di algoritmi di intelligenza artificiale

Si potrebbe pensare di implementare algoritmi di AI ai flussi di video, con lo scopo di ricavare informazioni sui video in riproduzione. Si potrebbe infatti applicare uno degli algoritmi, relativi all’intelligenza artificiale, citati nella parte di Background della tesi, in modo da poter estrarre dati, come ad esempio la qualità di esperienza dei vari video visualizzati, o classificare i video come appartenenti a determinati generi o in base alla fonte di provenienza, o ancora estrarre i momenti salienti del video in riproduzione. Come questi, esistono migliaia di possibili applicazioni di algoritmi di AI all’applicazione sviluppata.

4.2 Conclusioni

Grazie a questo studio siamo riusciti a saperne di più sullo streaming video e audio, sulla realtà estesa e su diverse tecniche di intelligenza artificiale utilizzate nello streaming di video. L'obiettivo di questo progetto era quello di sviluppare un'applicazione che collegasse la realtà estesa (nel mio caso, virtuale) all'interazione con flussi di video in streaming on-demand, permettendomi di sviluppare conoscenza pratica di questi argomenti. Siamo partiti dall'analisi di varia documentazione, per creare il capito di background della tesi, che ci ha permesso di acquisire conoscenze relative al concetto di realtà estesa e le sue varianti. Abbiamo poi condotto una breve digressione sui domini applicativi della realtà estesa, prima di passare ad analizzare alcune periferiche e alcuni dispositivi impiegati in questo campo. Abbiamo poi introdotto il concetto di streaming, seguito dai vari tipi di streaming e dal loro funzionamento. Ancora una volta, abbiamo fatto una breve digressione sui migliori servizi di streaming da utilizzare e sullo streaming pirata. L'introduzione dei concetti di AI, Machine Learning e Deep Learning, seguita da un'indagine su specifici algoritmi di AI utilizzati nel contesto dello streaming video, ha costituito la sezione finale della parte di background della tesi. Si è poi deciso di sviluppare una applicazione che ci permetesse di guardare video in streaming on demand in un ambiente virtuale, immersivo. L'applicazione è stata realizzata specificamente per Oculus Quest 2. La tesi comprende un capitolo che descrive l'applicazione creata, i suoi obiettivi e molti dettagli tecnici. Una sezione è dedicata all'architettura dell'applicazione e un'altra esamina le tecnologie e gli strumenti utilizzati. Come ultimo passo, è stata testata l'applicazione sviluppata su un piccolo gruppo di utenti per ottenere un loro feedback di utilizzo. Nella tesi, quindi, abbiamo inserito un paragrafo in cui discutiamo la valutazione preliminare dell'applicazione e analizziamo dei potenziali sviluppi futuri.

Ringraziamenti

Non avrei mai pensato di intraprendere un percorso universitario, altrettanto di concluderlo in questo lasso di tempo e in questo modo. Chi mi conosce sa quanto sia difficile per me riuscire ad esprimere le mie emozioni, ma mi sento in dovere di fare dei ringraziamenti per questo percorso di studi.

Un ringraziamento speciale va al mio relatore, il professore Fabio Palomba, per il supporto fornитоми durante il percorso di tirocincio, la stesura della tesi e per la grande disponibilità e professionalità che ha dimostrato nei miei confronti.

Ringrazio i miei genitori per avermi sostenuto e appoggiato nella scelta di questo percorso di studi ed essere stati presenti nei momenti di difficoltà e disponibili nei miei confronti.

Ringrazio il mio amico Mario, il mio punto di riferimento nei momenti di stallo che ho avuto durante la preparazione di qualche esame e che puntualmente si è armato di pazienza e mi ha dato una mano a superarli.

Ringrazio i miei amici di Gruppo Studio, che hanno affrontato, chi prima e chi dopo, questo percorso insieme a me e hanno resto questi tre anni indimenticabili. Avete sempre creduto in me e mi avete spronato a dare il massimo. Grazie per tutte le risate che abbiamo condiviso e per le ore di studio condivise insieme. Mi scuso con voi se a volte sono sembrato arrogante o spazientito.

Ringrazio il mio amico Antonio, e mio fratello Giuseppe per essere sempre stati presenti nella mia vita, per essersi sempre interessati al mio percorso di studi e avermi donato momenti di pausa dallo studio, grazie alle nostre pause caffè pomeridiane.

Ringrazio, Angela, Gennario, Giovanni e il piccolo Aniello, per avermi sempre stimato ed aver creduto in me, donandomi ulteriore forza per affrontare questo percorso.

Ringrazio la mia amata, Antonella, per avermi spronato a cominciare questo percorso, per esserci sempre stata, per aver ascoltato le mie (molto spesso inutili) paranoie e per aver passato parte del suo tempo ad insegnarmi matematica o ascoltarmi studiare. Sei stata per me forza e motivazione, accompagnandomi passo dopo passo al conseguimento di questo titolo. Grazie per aver capito e sopportato la mia, forse frequente, assenza dovuta allo studio.

Infine, un grazie speciale a me, per non aver mai mollato, essere stato forte e tenace ed aver dimostrato che nulla è impossibile se lo si vuole davvero.

Bibliografia

- [1] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Trans. Information Systems*, vol. vol. E77-D, no. 12, pp. 1321–1329, 12 1994. (Citato a pagina 4)
- [2] M. Billinghurst, A. Clark, and G. Lee, *A Survey of Augmented Reality*, vol. 8. 2015. (Citato a pagina 5)
- [3] J. Zheng, K. Chan, and I. Gibson, "Virtual reality," *Ieee Potentials*, vol. 17, no. 2, pp. 20–23, 1998. (Citato a pagina 7)
- [4] M. Speicher, B. D. Hall, and M. Nebeling, "What is mixed reality?," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–15, 2019. (Citato a pagina 7)
- [5] J. Krikke, "Streaming video transforms the media industry," *IEEE computer graphics and applications*, vol. 24, no. 4, pp. 6–12, 2004. (Citato a pagina 19)
- [6] pwc, "Pwc global entertainment & media outlook 2022-2026," 2022. (Citato a pagina 20)
- [7] Staff, "Il significato di streaming live e streaming on demand," 2020. (Citato alle pagine 22 e 25)
- [8] E. Fop, "Napster: La storia del software peer-to-peer che rivoluzionò la musica," 2017. (Citato a pagina 28)
- [9] P. N. Stuart J. Russell, *Intelligenza artificiale. Un approccio moderno. Ediz. mylab (Vol.)*. Pearson; 4° edizione. (Citato a pagina 29)
- [10] Z.-H. Zhou, *Machine learning*. Springer Nature, 2021. (Citato a pagina 30)

- [11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015. (Citato a pagina 31)
- [12] A. Shaout and B. Crispin, "Streaming video classification using machine learning.," *Int. Arab J. Inf. Technol.*, vol. 17, no. 4A, pp. 677–682, 2020. (Citato a pagina 33)
- [13] M. Ghosh, D. C. Singhal, and R. Wayal, "Desvq: Deep learning based streaming video qoe estimation," in *23rd International Conference on Distributed Computing and Networking*, pp. 19–25, 2022. (Citato a pagina 33)
- [14] H.-K. Han, Y.-C. Huang, and C. C. Chen, "A deep learning model for extracting live streaming video highlights using audience messages," in *Proceedings of the 2019 2nd Artificial Intelligence and Cloud Computing Conference*, pp. 75–81, 2019. (Citato a pagina 33)
- [15] Continisio, "Luci e ombre in unity," 2014. (Citato a pagina 53)
- [16] Continisio, "Baking e lightmaps," 2014. (Citato a pagina 55)
- [17] U. Technologies, "Skybox," 2017. (Citato a pagina 55)
- [18] Marzilli, "L'ambiente di sviluppo: Unity e visual studio," 2016. (Citato a pagina 56)
- [19] A. Hejlsberg, M. Torgersen, S. Wiltamuth, and P. Golde, *The C# programming language*. Pearson Education, 2008. (Citato a pagina 56)
- [20] T. Mullen, *Mastering blender*. John Wiley & Sons, 2011. (Citato a pagina 57)
- [21] U. Technologies, "Unitywebrequest," 2022. (Citato a pagina 58)
- [22] U. Technologies, "Oculus integration," 2022. (Citato a pagina 60)
- [23] Google, "Speech-to-text," (Citato a pagina 63)
- [24] Google, "Dialogflow," (Citato a pagina 64)