UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Informatica e dell'Automazione

Esperimenti di Machine Learning e Deep Learning nell'ambito del riconoscimento automatico di scene di violenza in flussi video.

Machine Learning and Deep Learning Experiments in the Context of Automatic Recognition of Violence Scenes in Video Streams.

Relatore:
Prof. Aldo Franco Dragoni

Relazione di:
Giulio Vignati

Correlatore:
Prof. Paolo Sernani

A.A. 2022/23

Abstract

La presente tesi si concentra sull'applicazione di reti neurali avanzate, nello specifico reti neurali convoluzionali ricorrenti (CNN-RNN) e modelli basati su trasformatori, per affrontare la sfida del riconoscimento automatico di scene di violenza in flussi video. L'obiettivo principale è sviluppare un sistema efficace e efficiente in grado di apprendere automaticamente le caratteristiche salienti delle scene violente e di riconoscerle con precisione.

Nel corso degli esperimenti, verranno addestrate e valutate diverse architetture di reti neurali, esplorando le potenzialità delle CNN-RNN nella cattura di informazioni spaziali e temporali, nonché l'efficacia dei modelli basati su trasformatori nel gestire relazioni complesse tra gli elementi della scena. L'addestramento sarà condotto utilizzando il dataset prodotto da AIRTLAB [1] [2].

Sommario

Elenco delle figure

Elenco delle tabelle

Introduzione

Nel tessuto digitale del mondo contemporaneo, siamo immersi in una proliferazione senza precedenti di dati visivi, veicolati attraverso flussi video provenienti da svariate fonti e piattaforme. Questa vasta quantità di informazioni, se da un lato rappresenta una ricchezza inestimabile, dall'altro pone notevoli sfide in termini di gestione e analisi. In questo contesto, emerge l'urgente necessità di sviluppare strumenti avanzati capaci di discernere e interpretare automaticamente i contenuti video, un compito reso sempre più cruciale dal crescente impiego di sistemi di videosorveglianza e dalla diffusione di materiale multimediale online.

La sicurezza è uno degli ambiti in cui l'applicazione di tecnologie innovative può produrre impatti significativi. Il riconoscimento automatico di scene di violenza in flussi video rappresenta un fronte strategico in questa direzione, con il potenziale di migliorare notevolmente la gestione delle emergenze, la prevenzione del crimine e la sicurezza pubblica. Attraverso l'impiego di tecniche avanzate di Machine Learning (ML) e Deep Learning (DL), è possibile affrontare la complessità dei dati visivi estrarre informazioni rilevanti per identificare situazioni di violenza.

Nel panorama odierno, caratterizzato da una vastità di informazioni e dalla crescente complessità delle minacce, il riconoscimento automatico di scene di violenza si configura come una risorsa preziosa. La capacità di analizzare e interpretare in tempo reale i flussi video può consentire una risposta più tempestiva a situazioni di emergenza, contribuendo così a migliorare la sicurezza delle comunità e degli spazi pubblici.

Le reti neurali sono al centro di queste innovazioni, e il campo del riconoscimento automatico di scene di violenza ha visto la proliferazione di architetture sofisticate come le reti neurali convoluzionali (CNN), reti neurali ricorrenti (RNN) e i modelli basati su trasformatori. Questi approcci combinano capacità di estrazione delle caratteristiche spaziali e temporali, permettendo una comprensione più profonda dei contenuti video.

L'obiettivo di questa tesi è esplorare in dettaglio l'efficacia di tali architetture nel contesto specifico del riconoscimento automatico di scene di violenza contribuendo così alla crescita del campo della sicurezza.

1.1 – Obiettivi della tesi

Questa tesi si propone di affrontare molteplici obiettivi nel contesto del riconoscimento automatico di scene di violenza nei flussi video, attraverso l'applicazione di approcci di machine learning e deep learning.

In primo luogo, l'obiettivo è condurre un'analisi approfondita delle tecnologie esistenti per il riconoscimento di scene violente. Ciò include una valutazione critica delle metodologie tradizionali e degli approcci basati su machine learning e deep learning. L'obiettivo è comprendere il panorama attuale, identificare le lacune esistenti e le sfide ancora da affrontare.

Successivamente, la tesi mira a sperimentare sui modelli di deep learning, concentrandosi in particolare su reti neurali convoluzionali (CNN), reti neurali ricorrenti (RNN) e modelli trasformatori. L'implementazione di tali modelli comporta l'addestramento su dataset specifici di scene violente per valutare le loro prestazioni e acquisire una comprensione più approfondita delle loro caratteristiche distintive.

Un altro obiettivo critico è la valutazione delle prestazioni dei modelli implementati attraverso metriche significative come accuratezza, recall, precisione e F1-score. L'obiettivo è fornire una valutazione completa delle capacità di riconoscimento delle scene di violenza dei modelli considerati.

1.2 – Applicazioni pratiche in contesti sociali

Il riconoscimento automatico di scene di violenza nei flussi video offre una serie di applicazioni pratiche di notevole rilevanza per la sicurezza pubblica e la gestione delle emergenze. L'utilizzo di tecnologie avanzate in questo contesto può produrre impatti significativi sulla prevenzione e risoluzione di situazioni critiche, fornendo alle autorità strumenti efficaci per garantire la sicurezza nelle aree pubbliche. Secondo i dati del Dipartimento di Giustizia degli Stati Uniti, le aggressioni fisiche nelle aree urbane sono aumentate del 12% nell'ultimo decennio, sottolineando l'importanza di affrontare questa problematica in modo proattivo [3].

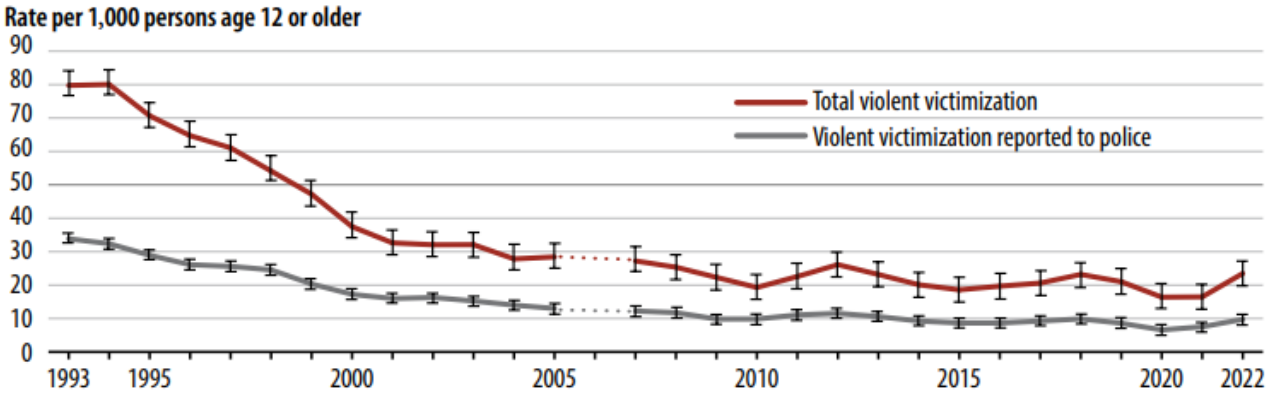

Tabella 1 - Tasso di vittimizzazione violenta e vittimizzazione violenta segnalata alla polizia, 1993-2022.

I dati raccolti attraverso diverse fonti, come le forze dell'ordine e le organizzazioni di sicurezza, possono essere integrati nei modelli di riconoscimento automatico per migliorare la precisione e l'efficacia degli algoritmi. L'analisi delle tendenze e dei pattern emergenti può contribuire a sviluppare strategie preventive mirate, rendendo possibile un approccio proattivo alla sicurezza pubblica.

In sintesi, l'applicazione di tecnologie di riconoscimento automatico di scene di violenza nei flussi video si presenta come una risorsa fondamentale per affrontare le sfide attuali della sicurezza pubblica, contribuendo a creare ambienti urbani più sicuri e reattivi.

Capitolo 2

Lo Stato dell'Arte

Il capitolo "Stato dell'arte" costituisce un'esplorazione approfondita del contesto attuale nel campo del riconoscimento automatico di scene di violenza nei flussi video mediante l'applicazione di tecnologie avanzate di machine learning e deep learning. Questa sezione mira a fornire una panoramica completa delle metodologie esistenti, degli sviluppi recenti e delle sfide ancora da affrontare in questo settore in continua evoluzione.

Attraverso una revisione critica della letteratura scientifica, il capitolo si propone di tracciare l'evoluzione delle tecnologie di riconoscimento di scene violente, partendo dalle approfondite metodologie tradizionali fino alle più avanzate e complesse architetture basate su reti neurali. Saranno esplorate le principali aree di ricerca, evidenziando i progressi significativi e mettendo in luce le lacune che richiedono ulteriori indagini.

Inoltre, il capitolo si concentrerà sull'analisi delle applicazioni pratiche di queste tecnologie, esplorando come il riconoscimento automatico di scene di violenza possa essere integrato in contesti come la videosorveglianza, la sicurezza pubblica e altri ambiti pertinenti. Si prenderanno in considerazione le sfide etiche e sociali connesse a tali applicazioni, enfatizzando l'importanza di un approccio equo e responsabile nell'implementazione di queste tecnologie.

Nel delineare lo stato attuale delle tecnologie di riconoscimento di scene violente, il capitolo mira a fornire una base solida per la comprensione delle sfide e delle opportunità nel campo, preparando il terreno per l'approfondimento delle metodologie sperimentali nell'ambito della presente tesi.

2.1 – Human Action Recognition

Il campo del riconoscimento delle azioni umane ha conosciuto notevoli progressi grazie all'evoluzione delle tecnologie di deep learning, contribuendo in maniera significativa a diverse applicazioni, come la sorveglianza video avanzata, l'analisi del comportamento umano e l'interazione uomo-macchina. Gli sviluppi recenti si sono concentrati su metodologie che sfruttano reti neurali avanzate, offrendo un aumento significativo delle prestazioni rispetto agli approcci tradizionali.

Le reti neurali convoluzionali (CNN) hanno giocato un ruolo chiave nel migliorare la capacità di riconoscimento delle azioni umane, permettendo una rappresentazione spaziale robusta delle caratteristiche visive nei video. Architetture come la Two-Stream CNN, introdotta da Simonyan e Zisserman, hanno dimostrato l'importanza di catturare simultaneamente informazioni spaziali e temporali per affrontare sfide complesse nel riconoscimento delle azioni umane [4].

Inoltre, l'adozione di reti neurali ricorrenti (RNN) e Long Short-Term Memory (LSTM) ha consentito una migliore gestione delle sequenze temporali, consentendo alle reti di catturare dinamiche temporali complesse presenti nei video. Studi come "Action Recognition with Improved Trajectories" di Wang et al. hanno dimostrato l'efficacia dell'utilizzo di LSTM nel riconoscimento di azioni umane, soprattutto quando si tratta di sequenze di lunga durata [5].

L'emergere delle reti basate su trasformatori ha ulteriormente avanzato il riconoscimento delle azioni umane, consentendo una gestione più efficace delle relazioni a lungo termine nei video. Modelli come l'Attentional Transformer, proposto da Carreira e Zisserman, hanno dimostrato la capacità di modellare interazioni spazio-temporali complesse nelle sequenze video, aprendo nuove prospettive per il riconoscimento delle azioni umane su larga scala [6].

In termini di framework di deep learning, Keras e TensorFlow continuano a rivestire un ruolo fondamentale nello sviluppo e nell'implementazione di modelli avanzati per il riconoscimento delle azioni umane. La loro flessibilità, facilità d'uso e scalabilità hanno contribuito alla diffusione di approcci innovativi in questo settore, consentendo a ricercatori e sviluppatori di esplorare nuove idee e migliorare costantemente le prestazioni dei modelli.

2.2 – Principali dataset di lavoro

L'analisi approfondita dei dataset di addestramento riveste un ruolo critico nel determinare il successo e l'efficacia degli algoritmi di riconoscimento automatico di scene di violenza nei flussi video. Nell'esplorare questo panorama ricco e diversificato, emergono diversi dataset chiave, ognuno progettato per affrontare specifiche sfide e scenari di utilizzo.

Il Moviescope Dataset si configura come un compendio eterogeneo di sequenze video, selezionate sia da produzioni cinematografiche che da contenuti multimediali online. Questa diversificazione è essenziale per l'addestramento di modelli in grado di comprendere le complesse dinamiche cinematografiche e, allo stesso tempo, affrontare le sfide uniche presenti nei flussi online, contribuendo così a migliorare la generalizzazione dei modelli.

Accanto a questo, l'AVA (Atomic Visual Actions) Dataset emerge come una risorsa fondamentale per il riconoscimento di azioni umane in sequenze video. Ciò che distingue questo dataset è la sua ricchezza di dettagli, con annotazioni che forniscono informazioni dettagliate su ogni frame. Questa caratteristica rende il dataset particolarmente adatto per l'analisi temporale delle dinamiche che portano agli episodi di violenza, migliorando la capacità dei modelli di cogliere i segnali chiave nel contesto temporale.

Una prospettiva più specifica è offerta dall'Hockey Fight Dataset, il quale si concentra esclusivamente sul riconoscimento di scene di violenza durante gli incontri di hockey. La peculiarità di questo dataset risiede nella sua capacità di catturare le particolarità delle situazioni violente in contesti sportivi specifici. L'inclusione di annotazioni temporali precise offre dettagli essenziali per analizzare le dinamiche degli scontri, fornendo un terreno di addestramento ricco di specificità contestuali.

Oltre a ciò, il UCF-Crime Dataset emerge come una risorsa dedicata al riconoscimento di attività criminali in video. Questo dataset è stato progettato specificamente per affrontare le sfide uniche legate al monitoraggio e alla prevenzione del crimine. La sua diversità di scenari, rappresentante attività criminali di varia natura, consente un addestramento mirato a contesti urbani e di sorveglianza, fornendo così una base solida per modelli in grado di cogliere la complessità delle attività criminali in video.

Ulteriori spunti emergono considerando il UCF-101 dataset, che si focalizza sul riconoscimento di azioni umane in video, coprendo una vasta gamma di categorie di attività. La sua eterogeneità

lo rende particolarmente adatto per addestrare modelli che devono affrontare molteplici contesti e comportamenti umani.

Infine, il MUG Facial Expression Database si distingue per il suo focus sul riconoscimento delle espressioni facciali, un aspetto cruciale per comprendere il contesto emotivo delle scene, specialmente in situazioni di potenziale violenza.

In conclusione, la selezione oculata di dataset di addestramento assume un ruolo centrale nella preparazione di modelli robusti per il riconoscimento automatico di scene di violenza. La diversità e la specificità di ciascun dataset contribuiscono a fornire ai modelli un bagaglio eterogeneo di esperienze, potenziando così la loro capacità di generalizzazione e comprensione di scenari realistici.

2.3 – Architettura delle reti neurali

Le reti neurali costituiscono una classe di modelli di apprendimento automatico che trae ispirazione dal funzionamento del cervello umano. La loro architettura complessa è fondamentale per la capacità di apprendere e rappresentare pattern complessi nei dati. Queste reti sono composte da neuroni artificiali, le unità fondamentali, che mimano in modo astratto il funzionamento dei neuroni biologici. Nel corso dell'apprendimento, la rete acquisisce la capacità di risolvere compiti complessi attraverso il rafforzamento o la diminuzione delle connessioni tra neuroni, modificando i pesi sinaptici.

La struttura delle reti neurali è organizzata in strati. Il layer di input riceve le informazioni iniziali, che vengono poi elaborate attraverso gli strati intermedi, noti come strati nascosti, e infine restituite come output dal layer di output. Gli strati intermedi contengono neuroni che apprendono rappresentazioni sempre più complesse e astratte dei dati in ingresso.

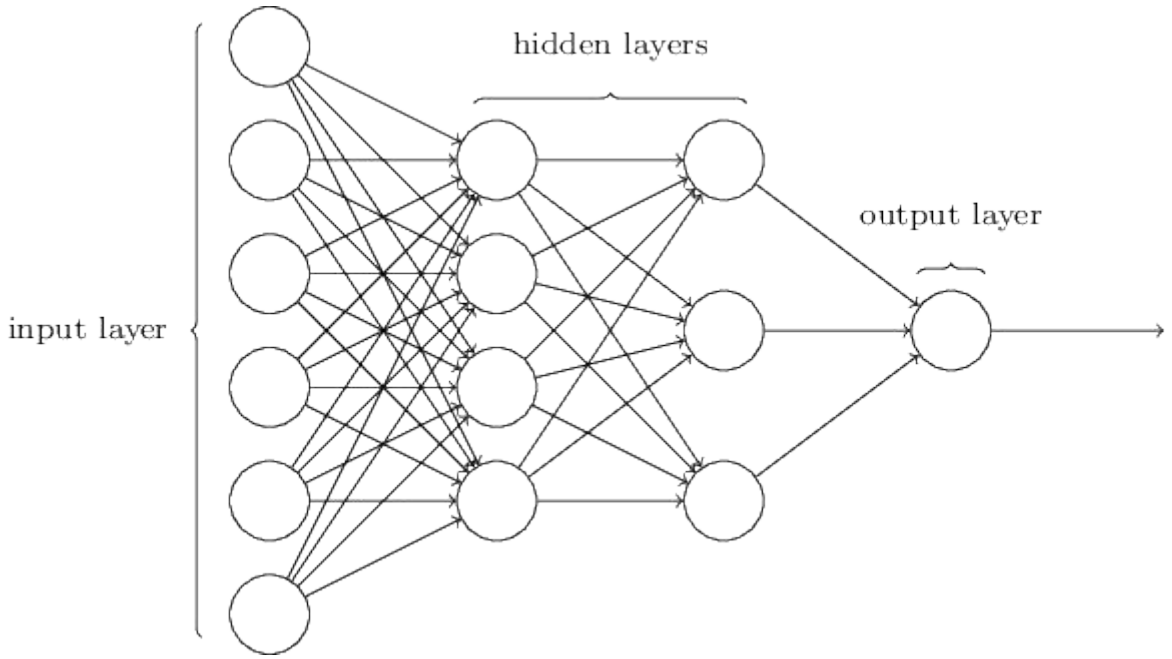

Figura 1 – Interazione tra i vari layer

Un aspetto fondamentale è la funzione di attivazione, che determina l'output di ciascun neurone in risposta alla somma ponderata dei suoi input. Questa funzione introduce la non-linearità, permettendo alla rete di modellare relazioni complesse nei dati. Tra le funzioni di attivazione comuni vi sono la sigmoide, che produce output compresi tra 0 e 1, e la ReLU (Rectified Linear Unit), che restituisce l'input se è positivo, altrimenti restituisce zero.

$$\sigma(z) \equiv \frac{1}{1+e^{-z}}.$$

Figura 2 – Formula matematica della sigmoide

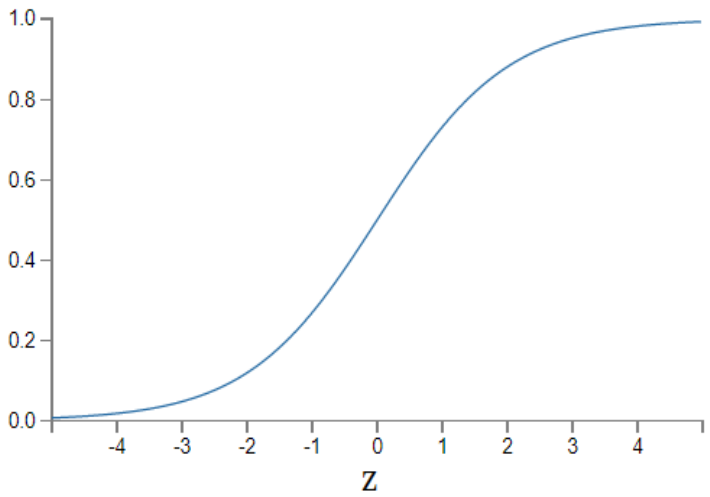

Figura 3 – Grafico della sigmoide

Durante l'addestramento, la rete cerca di minimizzare una funzione di perdita, che misura la discrepanza tra le sue previsioni e le etichette reali. Questo processo di ottimizzazione coinvolge la retropropagazione dell'errore, in cui il gradiente della funzione di perdita rispetto ai pesi delle connessioni viene calcolato e utilizzato per aggiornare i pesi in modo da ridurre la perdita complessiva.

La presenza di bias è un elemento chiave nell'adattabilità della rete. I bias sono valori aggiunti alle somme ponderate negli strati intermedi, fornendo una regolazione fine dell'output dei neuroni.

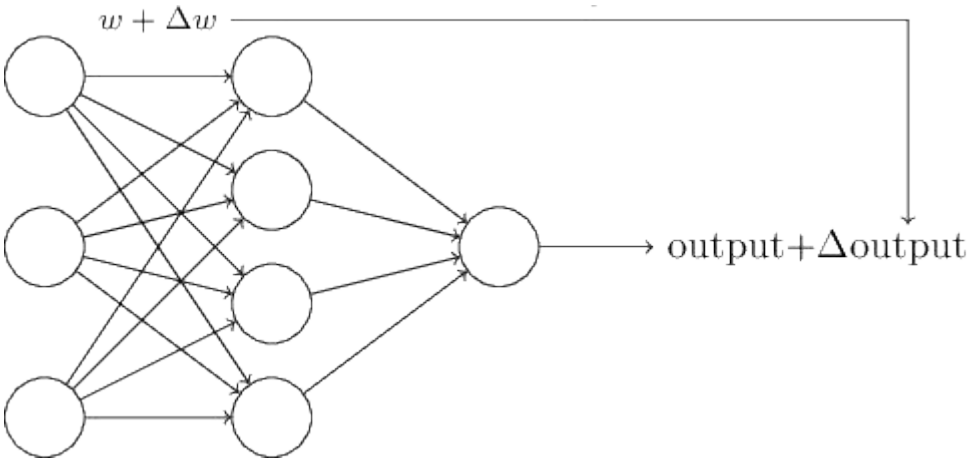

Figura 4 – Interazione tra bias e somme ponderate al raggiungimento di un output

Inoltre, la struttura delle reti neurali può essere influenzata dalla presenza di strati ricorrenti, che consentono alle informazioni di fluire ciclicamente attraverso la rete, utile per modellare sequenze e relazioni temporali.

In conclusione, l'architettura delle reti neurali si basa su principi bioispirati, con neuroni artificiali organizzati in strati e interconnessi mediante pesi sinaptici. Questa struttura, associata a funzioni di attivazione, processi di adattamento dei pesi e la presenza di bias, conferisce alle reti neurali la capacità di apprendere rappresentazioni complesse dai dati [7].

2.3.1 – Reti Neurali Convoluzionali (CNN)

Le reti neurali convoluzionali (CNN) rappresentano un vertice nell'evoluzione delle reti neurali, specializzandosi nell'analisi di dati multidimensionali come immagini e video. Introdotta per la prima volta da Yann LeCun e colleghi negli anni '90 [8], la struttura delle CNN è progettata per catturare pattern spaziali locali, rendendole particolarmente efficaci nel riconoscimento di pattern visivi.

Dal punto di vista architetturale, le CNN presentano uno schema di connessioni che sfrutta la convoluzione, la quale consiste nell'applicare filtri o kernel alle regioni sovrapposte dell'input per estrarre caratteristiche rilevanti. Questo processo di convoluzione riduce la complessità computazionale rispetto alle reti neurali fully connected, permettendo alle CNN di gestire input di grandi dimensioni in modo più efficiente.

Un elemento centrale delle CNN è il concetto di pooling, spesso eseguito attraverso strati di pooling, che riducono la dimensione delle rappresentazioni mantenendo le informazioni salienti. Ciò contribuisce a mantenere una rappresentazione spaziale delle features, consentendo alle CNN di apprendere pattern invarianti alle traslazioni.

I layer di convoluzione e pooling sono solitamente seguiti da uno o più strati fully connected, i quali integrano le informazioni estratte per generare un output finale. Durante l'addestramento, le CNN ottimizzano i pesi dei filtri attraverso la retropropagazione dell'errore, regolando la loro capacità di catturare features discriminanti.

Tra le architetture di punta, la LeNet-5 è stata una delle prime CNN ad ottenere successo nel riconoscimento di caratteri scritti a mano. Successivamente, AlexNet ha rivoluzionato il campo vincendo la ImageNet Large Scale Visual Recognition Challenge nel 2012, dimostrando la potenza delle CNN nelle applicazioni di visione artificiale [9].

La forza delle CNN risiede nella loro capacità di apprendere gerarchie di features, partendo da features di basso livello come bordi e texture, fino a features di alto livello che rappresentano concetti complessi. Questa struttura mirata alle immagini le ha consentito di ottenere risultati eccezionali in una vasta gamma di compiti, inclusi il riconoscimento di oggetti, il rilevamento di volti e la segmentazione semantica.

In conclusione, le reti neurali convoluzionali sono un pilastro nell'ambito della visione artificiale,

sfruttando la convoluzione e il pooling per catturare pattern spaziali nelle immagini. La loro architettura specializzata ha dimostrato successi significativi, evidenziando la loro efficacia nel riconoscimento di pattern visivi complessi.

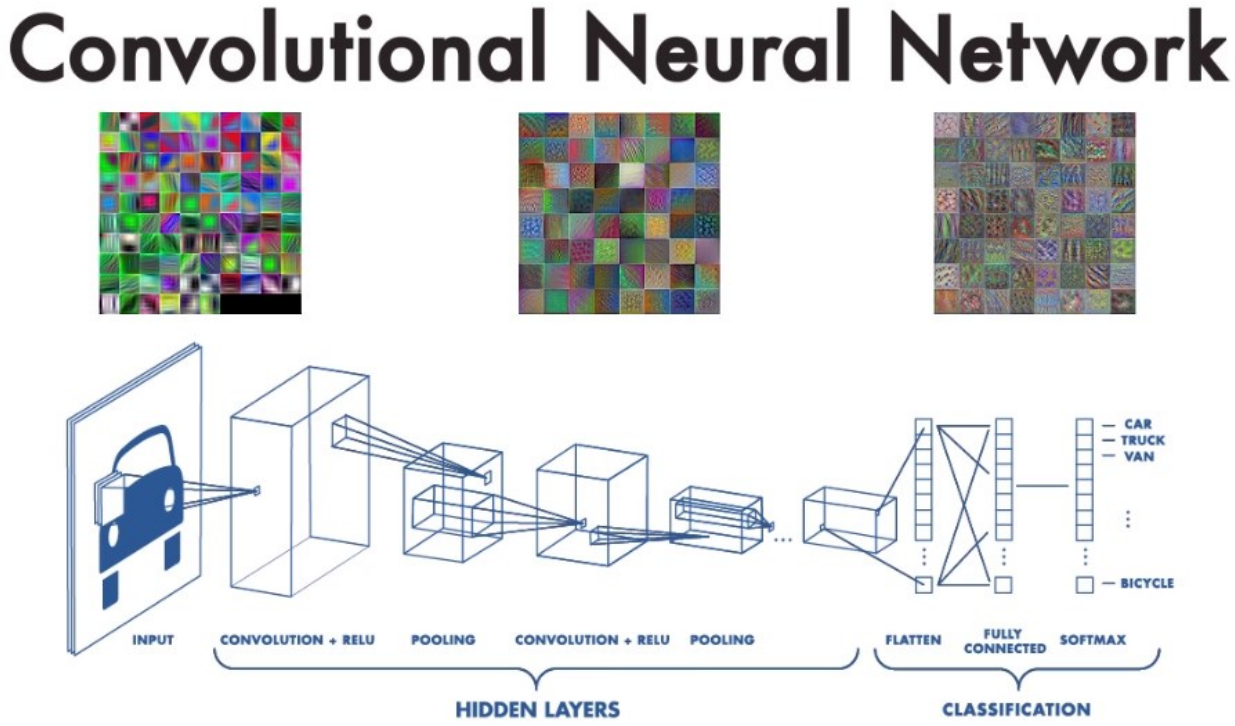

Figura 5 - Architettura di una rete CN

2.3.2 – Reti Neurali Ricorrenti (RNN)

Le reti neurali ricorrenti (RNN) costituiscono un ramo importante nell'evoluzione delle reti neurali, specializzandosi nell'analisi di dati sequenziali, temporali o di lunghezze variabili. A differenza delle reti neurali tradizionali, le RNN presentano connessioni cicliche tra i neuroni, consentendo loro di mantenere uno stato interno o memoria delle informazioni precedentemente elaborate.

L'architettura delle RNN è caratterizzata dalla presenza di unità ricorrenti, che permettono alle informazioni di essere propagate attraverso il tempo. Queste unità possiedono una struttura interna che incorpora il concetto di memoria, facilitando la conservazione di informazioni rilevanti per compiti che coinvolgono sequenze di dati.

Durante il passaggio dei dati attraverso una RNN, ogni unità ricorrente riceve input dallo stato interno precedente e dagli input correnti. L'output generato viene utilizzato sia come output finale che come input per la successiva iterazione, garantendo una retroazione temporale. Questo processo consente alle RNN di catturare dipendenze a lungo termine nelle sequenze, rendendole particolarmente adatte per compiti come il riconoscimento del linguaggio naturale, la traduzione automatica e la previsione temporale.

Tuttavia, le RNN presentano alcune limitazioni, come la difficoltà nel mantenere informazioni a lungo termine (problema del gradiente che svanisce o esplode). Per affrontare questo problema, sono state introdotte varianti di RNN più avanzate, come le reti LSTM (Long Short-Term Memory) e le reti GRU (Gated Recurrent Unit), che incorporano meccanismi di gating per gestire meglio le informazioni attraverso il tempo [11].

Le reti LSTM, proposte da Hochreiter e Schmidhuber nel 1997, sono dotate di una struttura più complessa rispetto alle RNN tradizionali. Introducono tre porte principali - porta di input, porta di output e porta di oblio - che regolano il flusso delle informazioni, migliorando la capacità di catturare dipendenze a lungo termine.

Le reti GRU [12] semplificano leggermente l'architettura delle LSTM, incorporando solo due porte principali - una porta di reset e una porta di aggiornamento. Le GRU sono in grado di ottenere prestazioni simili alle LSTM con una struttura più leggera.

In conclusione, le reti neurali ricorrenti costituiscono una classe fondamentale di modelli per il trattamento di dati sequenziali, grazie alla loro capacità di mantenere memoria attraverso il tempo. Varianti avanzate come le LSTM e le GRU hanno permesso di superare alcune limitazioni delle RNN tradizionali.

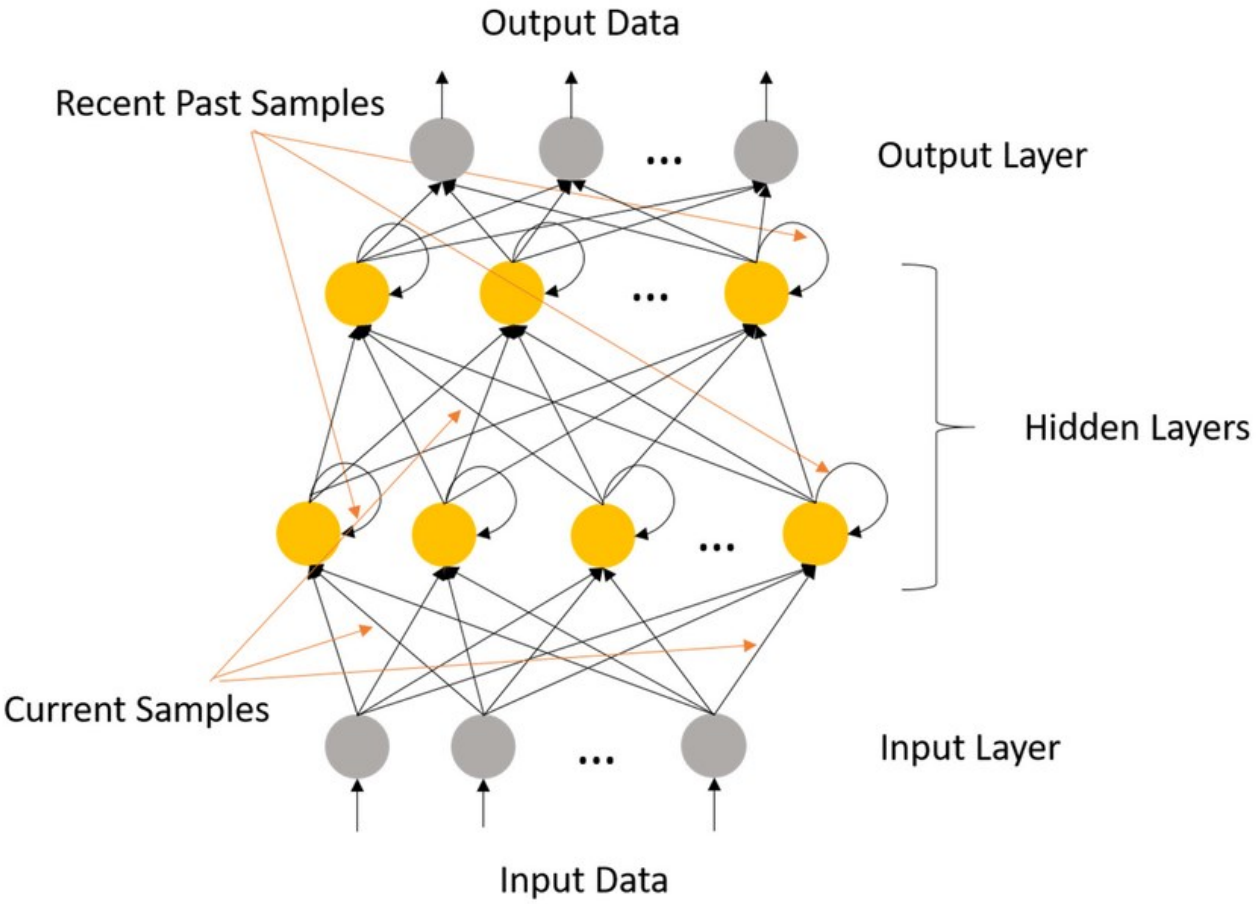

Figura 6 – Architettura di una rete RNN

2.3.3 – Reti Neurali Trasferibili (Transfer Learning)

Le reti neurali trasferibili, o reti neurali preaddestrate, rappresentano una metodologia avanzata nel campo del deep learning, mirata a risolvere sfide legate alla disponibilità di grandi quantità di dati di addestramento. Questo approccio si basa sull'idea di utilizzare una rete neurale già addestrata su un'ampia quantità di dati e trasferire le sue conoscenze acquisite ad un nuovo compito o dominio specifico.

L'architettura di una rete neurale trasferibile solitamente comprende diverse fasi. Inizialmente, una rete neurale viene addestrata su un compito particolare, spesso su un dataset massiccio e rappresentativo, per apprendere features generiche e di alto livello. Questa fase è chiamata preaddestramento.

Successivamente, la rete preaddestrata viene adattata o fine-tunata per il compito specifico di interesse. Questa fase coinvolge l'addestramento su un dataset più piccolo e orientato al nuovo compito. In questo modo, la rete è in grado di raffinare le sue rappresentazioni per adattarsi meglio alle caratteristiche specifiche del nuovo dominio.

Un esempio comune di rete neurale trasferibile è la famiglia di modelli chiamata "Inception," [14] che include l'InceptionV3, addestrato originariamente per il riconoscimento di immagini su ImageNet. Questi modelli hanno dimostrato di essere efficaci in molteplici compiti, dalla classificazione di immagini al rilevamento di oggetti.

Un altro esempio è rappresentato da modelli di linguaggio preaddestrati come BERT (Bidirectional Encoder Representations from Transformers) e GPT (Generative Pre-trained Transformer) [15]. Questi modelli, addestrati su enormi quantità di testi, sono successivamente adattati a compiti specifici di elaborazione del linguaggio naturale.

L'approccio trasferibile si è dimostrato particolarmente utile quando il dataset di addestramento per il compito di interesse è limitato o non completamente disponibile. Il trasferimento di conoscenze da compiti più ampi contribuisce alla costruzione di modelli più robusti e performanti.

In conclusione, le reti neurali trasferibili sfruttano la conoscenza preesistente acquisita da reti preaddestrate, adattandola in seguito a compiti specifici. Questa metodologia è cruciale in scenari in cui la disponibilità di dati di addestramento è limitata.

Capitolo 3

Guida all’esperimento

L'approfondimento successivo è dedicato alla disamina dettagliata dell'esperimento condotto, un'indagine che focalizza l'attenzione sull'applicazione di due approcci distinti, sfruttando avanzate architetture di reti neurali, al riconoscimento automatico di scene di violenza in flussi video. La realizzazione di due notebook su Google Colab ha permesso di sfruttare le potenzialità computazionali offerte dalla piattaforma, e la scelta di librerie di spicco come Keras e TensorFlow ha garantito un framework robusto per l'implementazione e l'addestramento dei modelli.

Il dataset utilizzato, gentilmente fornito da AIRTLAB Univpm [1][2], rappresenta un elemento cruciale di questo esperimento. La qualità e la diversità dei dati sono fondamentali per la validità delle analisi condotte, e la collaborazione con AIRTLAB ha permesso di accedere a risorse di alto livello per la realizzazione di questo esperimento.

Il primo notebook si concentra sull'implementazione di una rete CNN-RNN [16], combinando la potenza delle reti convoluzionali per l'estrazione di features spaziali con l'abilità delle reti ricorrenti nel catturare dinamiche temporali complesse. Tale approccio è stato selezionato per la sua comprovata efficacia in scenari di analisi sequenziale, come quello proposto dal riconoscimento di scene di violenza in sequenze video.

Il secondo notebook adotta invece un approccio basato sui trasformatori [17], una classe di modelli che ha recentemente dimostrato successi notevoli in varie applicazioni, inclusi problemi di elaborazione di sequenze visive.

Questo capitolo si propone di delineare il contesto, gli obiettivi e la metodologia dell'esperimento, fornendo una chiara comprensione delle scelte implementative e dei processi adottati. Successivamente, attraverso un'analisi approfondita dei risultati ottenuti da entrambi gli approcci, si mira a trarre conclusioni significative sull'efficacia delle reti neurali nel riconoscimento di scene di violenza nei video, con un'attenzione particolare alla comparazione tra le due architetture adottate.

3.1 – Costruzione del dataset

Il dataset utilizzato per condurre l'esperimento è un elemento chiave della ricerca, fornito gentilmente da AIRTLAB. La procedura di acquisizione inizia con l'utilizzo del comando *wget* per scaricare il dataset, seguito da una fase di decompressione al fine di rendere accessibili i suoi contenuti. Questo corpus comprende in totale 350 video, dei quali 310 sono destinati al training e 40 al test.

All'interno di ciascun video, sono presenti 175 scene riprese da due diverse camere, ciascuna angolata in modo differente per fornire una prospettiva variegata. La suddivisione tra video violenti e non violenti è chiaramente identificata all'interno del file .csv associato. Questa categorizzazione è fondamentale per addestrare e valutare l'efficacia delle reti neurali nel riconoscimento di scene di violenza.

Per quanto riguarda il processo di addestramento, è stato adottato un approccio bilanciato, in cui entrambe le reti neurali hanno beneficiato della stessa porzione di dataset. La possibilità di eseguire il training direttamente, con un impegno temporale di circa 20 minuti, è stata resa disponibile. Allo stesso tempo, per ottimizzare il tempo, è stata offerta l'opzione di scaricare un modello preaddestrato.

È fondamentale sottolineare la brevità delle scene, le quali, per evitare sovraccarichi computazionali e incorporare solo informazioni essenziali, sono state limitate a pochi secondi. Da ciascuna di queste, viene estratto un insieme di 20 frame per analizzare la dinamica della scena. Partendo dal frame centrale, vengono selezionati i 10 frame precedenti e i 10 successivi, consentendo l'estrazione delle parti rilevanti e rafforzando la capacità delle reti neurali di riconoscere pattern temporali significativi.

Questa scelta metodologica è stata motivata dalla necessità di bilanciare l'efficienza computazionale con la precisione del modello. La strategia di selezione dei frame mira a concentrarsi sulle porzioni più rilevanti delle scene, mitigando il rischio di sovraccaricare la rete con informazioni non cruciali.

In conclusione, il dataset rappresenta un pilastro fondamentale dell'esperimento, fornendo la base per l'addestramento e la valutazione delle reti neurali. La sua eterogeneità, insieme alle scelte ponderate riguardanti la durata delle scene e la selezione dei frame, contribuisce a garantire la robustezza e la significatività delle analisi condotte nell'ambito di questa ricerca universitari.

3.2 – Training delle reti

Il capitolo dedicato al training delle reti è cruciale per comprendere come i modelli di deep learning acquisiscano conoscenza dai dati forniti. In questo contesto, esploreremo il processo di addestramento sia per le reti CNN-RNN che per quelle basate su trasformatori, focalizzandoci sugli aspetti distintivi di ciascuna architettura.

Il training delle reti CNN-RNN inizia con l'alimentazione dei dati nel modello. In questo caso, i frame estratti dalle brevi scene di video vengono presentati alla rete. La componente convoluzionale della rete, composta da strati convoluzionali e di pooling, estrae features spaziali rilevanti da ciascun frame. Successivamente, la sequenza di features estratte viene fornita alla parte ricorrente della rete, solitamente basata su strati LSTM o GRU. Questa fase consente alla rete di catturare relazioni temporali e modelli di lungo termine nei dati. Durante il training, i pesi delle connessioni della rete vengono aggiornati attraverso l'ottimizzazione della funzione di perdita. Questa funzione misura la discrepanza tra le previsioni del modello e le etichette reali associate ai dati di addestramento. L'uso di tecniche come la retropropagazione e la discesa del gradiente permette di regolare gradualmente i pesi per migliorare le performance del modello.

Nel caso delle reti basate su trasformatori, l'addestramento segue un approccio leggermente diverso. L'architettura trasformatoriale elimina la necessità di strati ricorrenti, enfatizzando l'attenzione su meccanismi di autoattenzione. Durante il training, la sequenza di frame viene processata attraverso strati multihead self-attention. Questi strati attribuiscono pesi differenti ai vari frame in base alla loro rilevanza per il contesto, consentendo al modello di focalizzarsi su particolari parti della sequenza. Analogamente alle CNN-RNN, l'addestramento si basa sull'ottimizzazione della funzione di perdita. La differenza chiave risiede nella gestione dell'attenzione e nell'apprendimento dei pesi attraverso il meccanismo di attenzione, che può facilitare il riconoscimento di pattern temporali complessi. In entrambi i casi, è possibile sfruttare tecniche di trasferimento di apprendimento, utilizzando modelli preaddestrati su dataset di immagini più ampi. Questo approccio può accelerare il processo di addestramento e migliorare le performance, specialmente in scenari con quantità limitate di dati di addestramento. In conclusione, il capitolo di training delle reti si focalizza sulla convergenza dei modelli verso rappresentazioni significative dei dati, evidenziando le specificità e le differenze tra le reti CNN-RNN e quelle basate su trasformatori. La scelta dell'architettura dipende dall'applicazione specifica e dalla natura dei dati trattati. Il training delle reti CNN-RNN inizia con l'alimentazione dei dati nel modello. In questo caso, i frame estratti dalle brevi scene di video vengono presentati

alla rete. La componente convoluzionale della rete, composta da strati convoluzionali e di pooling, estrae features spaziali rilevanti da ciascun frame.

Di seguito, è illustrato come funziona il label di un video. Il file .csv ha due dati: il nome del video, costruito in maniera tale da avere il numero progressivo del video, la sigla "nv" o "v" per indicare rispettivamente "non-violent" o "violent", il numero 1 o 2 che indica la camera utilizzata. Il tag è un semplice numero binario uguale a zero nel caso di un video non violento, e uguale a 1 nel caso di un video violento.

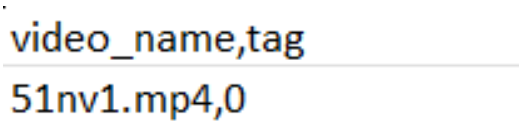

Figura 7 – Esempio di label.

3.3 – Esperimento su rete CNN-RNN

Dopo la fase di addestramento, vengono eseguite una serie di operazioni cruciali per valutare le performance del modello di sequenza. L'implementazione si basa su una struttura di rete neurale che utilizza uno strato GRU (Gated Recurrent Unit) per catturare le dinamiche temporali delle sequenze di frame. Questa architettura è definita mediante la creazione di un modello che accetta input relativi alle features estratte dai frame e una maschera booleana, che gioca un ruolo essenziale nel considerare solo le posizioni rilevanti durante l'elaborazione della sequenza.

Le epoche, un concetto centrale nell'addestramento, rappresentano il numero di passaggi completati attraverso l'intero dataset durante il training. Ciascuna epoca consente al modello di apprendere dai dati, aggiornando i pesi delle connessioni tra gli strati della rete. L'utilizzo del concetto di epoche è fondamentale per determinare la quantità di informazioni che il modello ha avuto l'opportunità di assimilare durante il processo di training.

I weights, o pesi delle connessioni, sono parametri fondamentali di una rete neurale. Essi rappresentano l'importanza relativa di ciascuna connessione tra i neuroni. Nel contesto dell'operazione di checkpoint, l'attributo *save_weights_only* indica che verranno salvati solo i pesi del modello, preservando l'architettura del modello stesso. Questo è particolarmente utile per il ripristino e la condivisione dei pesi addestrati in situazioni successive senza dover memorizzare l'intero modello.

L'utilizzo degli strati è una componente chiave dell'architettura della rete. In questo caso, la struttura è composta da uno strato GRU seguito da uno strato di dropout per mitigare il rischio di overfitting e uno strato denso che produce le previsioni finali. Ogni strato svolge un ruolo specifico nel processo di apprendimento e contribuisce alla complessità e alle capacità del modello nel comprendere e generalizzare dai dati di input.

L'early stopping, implementato attraverso l'oggetto *tf.keras.callbacks.EarlyStopping*, è una pratica significativa durante il training. Questo meccanismo monitora la funzione di perdita sulla parte di validazione e interrompe il training quando la perdita non migliora dopo un numero prestabilito di epoche, contribuendo a evitare l'overfitting e garantendo che il modello conservi una buona capacità di generalizzazione.

In sintesi, l'esecuzione di questo blocco di codice implica il caricamento dei pesi preaddestrati, la definizione del modello di sequenza, il training attraverso multiple epoche con monitoraggio della

performance attraverso la funzione di perdita, l'applicazione dell'early stopping per ottimizzare le performance, e infine la valutazione del modello sui dati di test per determinare l'accuratezza finale del sistema.

3.3.1 – Risultati raggiunti

L'analisi dei risultati rappresenta un momento cruciale per valutare l'efficacia del modello implementato nel contesto del riconoscimento di scene di violenza nei video. I test sono stati condotti sull'intera cartella di video di test, e i risultati sono presentati in una tabella che include il nome del video, l'azione corretta e la percentuale di rilevazione, specificando se l'azione rilevata è stata classificata come 0 (non violenta) o 1 (violenta), accompagnata dalla relativa percentuale di confidenza.

La valutazione dei risultati richiede una considerazione attenta delle metriche di performance del modello. La corretta identificazione delle azioni, rispecchiate nella colonna "Azione corretta", è fondamentale per valutare la capacità del modello di discriminare tra comportamenti violenti e non violenti. La corrispondenza tra l'azione prevista dal modello e l'azione reale nei video è un indicatore chiave delle performance generali.

La percentuale di rilevazione rappresenta la fiducia del modello nella sua previsione, espressa come percentuale di probabilità. Questo aspetto è particolarmente rilevante in scenari in cui la certezza della previsione può influenzare le decisioni di sistema. Ad esempio, una percentuale di rilevazione più alta potrebbe suggerire una maggiore confidenza del modello nella correttezza della sua previsione.

L'analisi dei risultati dovrebbe anche considerare possibili casi di falsi positivi o falsi negativi. Un falso positivo si verifica quando il modello classifica erroneamente un'azione come violenta quando non lo è, mentre un falso negativo si verifica quando un'azione violenta non viene correttamente identificata dal modello. Questi errori possono fornire insight sulla sensibilità e specificità del modello rispetto al problema specifico affrontato.

Inoltre, sarebbe utile esaminare la distribuzione delle percentuali di rilevazione per comprendere come il modello assegna confidenza alle sue previsioni. Ad esempio, potrebbe essere interessante valutare se il modello tende a essere più conservativo assegnando percentuali di rilevazione più basse o se mostra una tendenza a essere più audace nel fornire previsioni con percentuali elevate.

Infine, l'analisi dei risultati dovrebbe concludersi con una riflessione critica sulle potenziali aree di miglioramento del modello. Ciò potrebbe includere considerazioni sull'aggiunta di nuovi dati di addestramento, l'ottimizzazione degli iperparametri o l'esplorazione di architetture di reti neurali alternative.

Complessivamente, ci possiamo ritenere soddisfatti di questo modello, in quanto la sua accuratezza, dato con la quale si verifica la percentuale di azione identificate correttamente, ha raggiunto il valore dell'85%.

Nome video	Azione corretta	Azione rilevata tramite modello CNN-RNN	
51nv1.mp4	nonviolent	nonviolent 56.62%	violent 43.38%
52nv1.mp4	nonviolent	nonviolent 66.98%	violent 33.02%
53nv1.mp4	nonviolent	violent 51.04%	nonviolent 48.96%
54nv1.mp4	nonviolent	nonviolent 68.03%	violent 31.97%
55nv1.mp4	nonviolent	nonviolent 50.63%	violent 49.37%
56nv1.mp4	nonviolent	violent 57.38%	nonviolent 42.62%
57nv1.mp4	nonviolent	nonviolent 50.34%	violent 49.66%
58nv1.mp4	nonviolent	nonviolent 64.39%	violent 35.61%
59nv1.mp4	nonviolent	violent 75.08%	nonviolent 24.92%
60nv1.mp4	nonviolent	violent 75.34%	nonviolent 24.66%
51nv2.mp4	nonviolent	nonviolent 53.87%	violent 46.13%
52nv2.mp4	nonviolent	nonviolent 66.40%	violent 33.60%
53nv2.mp4	nonviolent	nonviolent 69.30%	violent 30.70%
54nv2.mp4	nonviolent	nonviolent 69.25%	violent 30.75%
55nv2.mp4	nonviolent	nonviolent 71.63%	violent 28.37%
56nv2.mp4	nonviolent	nonviolent 66.94%	violent 33.06%
57nv2.mp4	nonviolent	nonviolent 67.37%	violent 32.63%
58nv2.mp4	nonviolent	nonviolent 66.24%	violent 33.76%
59nv2.mp4	nonviolent	nonviolent 66.04%	violent 33.96%
60nv2.mp4	nonviolent	nonviolent 70.54%	violent 29.46%
106v1.mp4	violent	violent 78.14%	nonviolent 21.86%
107v1.mp4	violent	violent 81.56%	nonviolent 18.44%
108v1.mp4	violent	violent 84.60%	nonviolent 15.40%
109v1.mp4	violent	violent 83.58%	nonviolent 16.42%
110v1.mp4	violent	violent 76.35%	nonviolent 23.65%
111v1.mp4	violent	violent 61.50%	nonviolent 38.50%
112v1.mp4	violent	violent 81.32%	nonviolent 18.68%
113v1.mp4	violent	violent 80.52%	nonviolent 19.48%
114v1.mp4	violent	violent 83.33%	nonviolent 16.67%
115v1.mp4	violent	violent 53.62%	nonviolent 46.38%
106v2.mp4	violent	violent 54.35%	nonviolent 45.65%
107v2.mp4	violent	violent 61.47%	nonviolent 38.53%
108v2.mp4	violent	violent 75.85%	nonviolent 24.15%
109v2.mp4	violent	violent 50.39%	nonviolent 49.61%
110v2.mp4	violent	violent 58.23%	nonviolent 41.77%
111v2.mp4	violent	nonviolent 68.02%	violent 31.98%
112v2.mp4	violent	nonviolent 51.68%	violent 48.32%
113v2.mp4	violent	violent 66.32%	nonviolent 33.68%
114v2.mp4	violent	violent 74.39%	nonviolent 25.61%
115v2.mp4	violent	violent 58.31%	nonviolent 41.69%

Figura 8 – Risulati raggiunti dalla rete CNN-RNN

3.3.2 – Analisi del modello CNN-RNN

Il metodo *model.summary()* [18] è un prezioso strumento per esplorare l'architettura di un modello CNN-RNN. Questa chiamata offre un dettagliato resoconto degli strati del modello, evidenziando il numero di parametri totali, addestrabili e non addestrabili. Approfondiamo ulteriormente questi aspetti per ottenere una comprensione più approfondita.

I parametri totali rappresentano l'insieme completo delle variabili che costituiscono il modello, inclusi pesi e bias. Questi sono fondamentali per l'apprendimento del modello dai dati durante il processo di addestramento.

I parametri addestrabili, invece, indicano il numero di variabili che il modello può regolare durante l'addestramento.

In contrasto, i parametri non addestrabili rimangono costanti durante il processo di addestramento. Questi sono spesso associati a strati o operazioni che non richiedono aggiornamenti durante l'ottimizzazione. Ad esempio, uno strato di pooling esegue un'operazione fissa senza bisogno di apprendimento, rendendo i suoi parametri non addestrabili.

L'analisi di *model.summary()* fornisce una panoramica approfondita della complessità e della capacità di apprendimento del modello. Un elevato numero di parametri suggerisce una maggiore flessibilità del modello nell'adattarsi ai dati di addestramento, ma può comportare il rischio di overfitting. Trovare un equilibrio tra complessità e generalizzazione è cruciale per sviluppare modelli efficienti e robusti.

```
Model: "model"
_________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #     Connected to
=================================================================================================
 input_3 (InputLayer)        [(None, 20, 2048)]        0           []

 input_4 (InputLayer)        [(None, 20)]              0           []

 gru (GRU)                   (None, 20, 16)            99168       ['input_3[0][0]',
                                                                    'input_4[0][0]']

 gru_1 (GRU)                 (None, 8)                 624         ['gru[0][0]']

 dropout (Dropout)           (None, 8)                 0           ['gru_1[0][0]']

 dense (Dense)               (None, 8)                 72          ['dropout[0][0]']

 dense_1 (Dense)             (None, 2)                 18          ['dense[0][0]']

=================================================================================================
Total params: 99,882
Trainable params: 99,882
Non-trainable params: 0
```

Figura 9 - Model.summary() del modello CNN-RNN.

3.4 – Esperimento su rete Transformers

Dopo il processo di addestramento del modello, vengono eseguite diverse operazioni per consolidare e valutare le prestazioni del modello allenato.

Innanzitutto, è presente una classe personalizzata chiamata *PositionalEmbedding.* Questa classe implementa l'embedding posizionale, un componente cruciale nei modelli di tipo trasformatore. L'embedding posizionale aggiunge informazioni sulla posizione relativa delle diverse parti della sequenza di input, contribuendo a catturare relazioni temporali nei dati. La sua implementazione coinvolge l'utilizzo di un layer di embedding e l'aggiunta di queste posizioni embeddate agli input originali.

Successivamente, c'è la classe *TransformerEncoder*, che rappresenta uno strato di encoding nel modello trasformatore. Questo strato applica l'attenzione multi-testa seguita da una proiezione densa e normalizzazione layer-wise. L'attenzione multi-testa consente al modello di concentrarsi su diverse parti della sequenza in modo simultaneo, mentre la proiezione densa contribuisce a modellare le relazioni complesse nei dati.

La funzione *get_compiled_model* compone l'architettura complessiva del modello utilizzando le classi precedentemente definite. Il modello comprende un embedding posizionale, uno strato trasformatore e strati di pooling e dropout per la riduzione dimensionale e la regolarizzazione. La compilazione del modello coinvolge la definizione dell'ottimizzatore, della funzione di perdita e delle metriche di valutazione.

La funzione *run_experiment* gestisce l'intero processo di addestramento del modello. Durante l'addestramento, vengono utilizzati callback, come *ModelCheckpoint*, che salva i pesi migliori del modello. Questo è particolarmente utile per evitare il sovradattamento ai dati di addestramento.

Infine, il modello viene valutato utilizzando il set di test, e la sua accuratezza viene stampata a scopo informativo. L'utilizzo dei pesi salvati precedentemente durante il miglior epoca consente di ottenere il modello ottimale per la generalizzazione.

In generale, questo processo illustra come l'embedding posizionale e gli strati trasformatori vengano integrati in un modello più ampio. L'addestramento progressivo del modello attraverso le epoche, unitamente all'ottimizzazione dei pesi, contribuisce al miglioramento delle capacità predittive del modello.

3.4.1 – Risultati raggiunti

L'analisi dei risultati del modello Transformers rivela una situazione in cui le prestazioni del modello non soddisfano le aspettative, poiché non riesce a riconoscere correttamente i filmati. Questa sfida può essere attribuita a diversi fattori, tra cui la complessità intrinseca del modello rispetto al problema specifico di riconoscimento di scene di violenza nei filmati.

La complessità dei modelli trasformatori, pur essendo potente in molte applicazioni, potrebbe risultare eccessiva per la natura specifica di questo compito. La dimensione limitata del dataset potrebbe limitare la capacità del modello di apprendere rappresentazioni significative delle scene di violenza. Una raccolta più ampia e diversificata di dati potrebbe essere necessaria per una migliore generalizzazione.

Inoltre, la scelta degli iperparametri potrebbe non essere ottimale. Parametri come il tasso di apprendimento, il numero di teste nell'attenzione multi-testa e le dimensioni degli strati devono essere attentamente ottimizzati per migliorare le prestazioni del modello. Problemi nella normalizzazione dei dati e il rischio di overfitting potrebbero ulteriormente influenzare il rendimento del modello.

Un aspetto critico potrebbe essere la mancanza di diversità nelle scene di violenza nei filmati, limitando la capacità del modello di apprendere in modo robusto. Introdurre una maggiore varietà nei dati di addestramento potrebbe rivelarsi essenziale. Infine, il modello potrebbe avere difficoltà a comprendere il contesto delle scene di violenza all'interno di un filmato, suggerendo la necessità di considerazioni temporali o informazioni contestuali aggiuntive.

In sintesi, la mancata capacità del modello Transformers di riconoscere i filmati può essere una conseguenza della complessità, delle dimensioni del dataset, della configurazione degli iperparametri e della mancanza di diversità nei dati di addestramento. Affrontare questi aspetti richiede un'analisi approfondita e una continua iterazione per migliorare il modello e le sue capacità di generalizzazione.

Nome video	Azione corretta	Azione rilevata tramite modello Transofrmers	
51nv1.mp4	nonviolent	nonviolent: 84.92%	violent: 15.08%
52nv1.mp4	nonviolent	nonviolent: 82.15%	violent: 17.85%
53nv1.mp4	nonviolent	nonviolent: 79.05%	violent: 20.95%
54nv1.mp4	nonviolent	nonviolent: 79.33%	violent: 20.67%
55nv1.mp4	nonviolent	nonviolent: 82.44%	violent: 17.56%
56nv1.mp4	nonviolent	nonviolent: 82.01%	violent: 17.99%
57nv1.mp4	nonviolent	nonviolent: 80.58%	violent: 19.42%
58nv1.mp4	nonviolent	nonviolent: 82.31%	violent: 17.69%
59nv1.mp4	nonviolent	nonviolent: 82.23%	violent: 17.77%
60nv1.mp4	nonviolent	nonviolent: 83.65%	violent: 16.35%
51nv2.mp4	nonviolent	nonviolent: 79.98%	violent: 20.02%
52nv2.mp4	nonviolent	nonviolent: 78.96%	violent: 21.04%
53nv2.mp4	nonviolent	nonviolent: 81.72%	violent: 18.28%
54nv2.mp4	nonviolent	nonviolent: 81.87%	violent: 18.13%
55nv2.mp4	nonviolent	nonviolent: 79.92%	violent: 20.08%
56nv2.mp4	nonviolent	nonviolent: 78.44%	violent: 21.56%
57nv2.mp4	nonviolent	nonviolent: 85.30%	violent: 14.70%
58nv2.mp4	nonviolent	nonviolent: 81.92%	violent: 18.08%
59nv2.mp4	nonviolent	nonviolent: 78.68%	violent: 21.32%
60nv2.mp4	nonviolent	nonviolent: 84.14%	violent: 15.86%
106v1.mp4	violent	nonviolent: 80.79%	violent: 19.21%
107v1.mp4	violent	nonviolent: 82.35%	violent: 17.65%
108v1.mp4	violent	nonviolent: 84.74%	violent: 15.26%
109v1.mp4	violent	nonviolent: 81.56%	violent: 18.44%
110v1.mp4	violent	nonviolent: 83.57%	violent: 16.43%
111v1.mp4	violent	nonviolent: 80.81%	violent: 19.19%
112v1.mp4	violent	nonviolent: 81.30%	violent: 18.70%
113v1.mp4	violent	nonviolent: 79.94%	violent: 20.06%
114v1.mp4	violent	nonviolent: 86.11%	violent: 13.89%
115v1.mp4	violent	nonviolent: 79.53%	violent: 20.47%
106v2.mp4	violent	nonviolent: 84.14%	violent: 15.86%
107v2.mp4	violent	nonviolent: 83.13%	violent: 16.87%
108v2.mp4	violent	nonviolent: 86.90%	violent: 13.10%
109v2.mp4	violent	nonviolent: 85.31%	violent: 14.69%
110v2.mp4	violent	nonviolent: 87.88%	violent: 12.12%
111v2.mp4	violent	nonviolent: 83.67%	violent: 16.33%
112v2.mp4	violent	nonviolent: 84.98%	violent: 15.02%
113v2.mp4	violent	nonviolent: 81.17%	violent: 18.83%
114v2.mp4	violent	nonviolent: 83.64%	violent: 16.36%
115v2.mp4	violent	nonviolent: 87.24%	violent: 12.76%

Figura 10 - Risultati raggiunti dal modello Transformers.

3.4.2 – Analisi del modello Transformers

Come per la rete precedente, ci soffermeremo alcuni istanti ad analizzare i parametri del nostro modello.

In un modello Transformers, la complessità deriva dalla presenza di numerosi parametri dovuti agli strati di attenzione multi-testa e alle proiezioni dense. Contrariamente al modello CNN-RNN precedentemente considerato, il modello Transformers presenta un numero significativamente più elevato di parametri totali, che possono arrivare a milioni.

Nel caso specifico, la disparità di parametri totali tra il modello CNN-RNN e quello Transformers (100,000 rispetto a 16 milioni) è significativa. Questo divario notevole potrebbe, in effetti, contribuire alle difficoltà del modello Transformers nel convergere a risultati decenti. La complessità elevata potrebbe portare a una maggiore suscettibilità all'overfitting, rendendo il modello più sensibile ai dati di addestramento e meno in grado di generalizzare su nuovi dati.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_3 (InputLayer)        [(None, None, None)]      0

 frame_position_embedding (P  (None, None, 2048)       40960
 ositionalEmbedding)

 transformer_layer (Transfor  (None, None, 2048)       16812036
 merEncoder)

 global_max_pooling1d (Globa  (None, 2048)             0
 lMaxPooling1D)

 dropout (Dropout)           (None, 2048)              0

 dense_2 (Dense)             (None, 2)                 4098

=================================================================
Total params: 16,857,094
Trainable params: 16,857,094
Non-trainable params: 0
```

Figura 11 – Model.summary() del modello Transformers.

Capitolo 4

Conclusioni e sviluppi futuri

In conclusione, l'esplorazione dei modelli di apprendimento profondo, inclusi la rete CNN-RNN e Transformers, nel contesto del riconoscimento automatico di scene di violenza in flussi video, ha rivelato una serie di sfide e opportunità che meritano attenta considerazione.

L'adozione di modelli neurali avanzati, come i Transformers, ha portato ad un aumento significativo della complessità, manifestato da un notevole incremento dei parametri totali rispetto alla rete CNN-RNN. Questa complessità, sebbene promettente per applicazioni sofisticate, potrebbe rappresentare uno dei motivi alla base delle divergenze nei risultati ottenuti tra i due modelli.

Il divario sostanziale nei parametri totali ha sollevato interrogativi sulla capacità del modello Transformers di adattarsi in maniera efficiente ai dati di addestramento, suggerendo che la sua elevata complessità potrebbe contribuire alle difficoltà riscontrate nella fase di apprendimento e convergenza.

L'importanza del dataset è emersa chiaramente, evidenziando la necessità di raccolte dati ampie e diversificate per garantire una rappresentazione completa delle scene di violenza. La mancanza di diversità nei dati potrebbe essere un fattore determinante nella disparità di risultati, influenzando la capacità del modello di apprendere pattern robusti e di generalizzare su nuovi contesti.

I risultati sperimentali hanno sottolineato la rilevanza dell'ottimizzazione degli iperparametri, suggerendo che un bilanciamento accurato tra la complessità del modello e la dimensione del dataset è cruciale per ottenere prestazioni ottimali.

Per prospettive future, potrebbe essere vantaggioso concentrarsi sul raffinamento delle architetture dei modelli, l'indagine di tecniche di regolarizzazione più efficaci e l'acquisizione di dataset più ampi e diversificati. Inoltre, esplorare collaborazioni tra esperti di dominio e specialisti di machine learning potrebbe favorire l'integrazione di conoscenze contestuali nel processo di addestramento, migliorando così la comprensione del contesto nelle scene di violenza e potenzialmente riducendo le disparità nei risultati tra i modelli analizzati.

In definitiva, il cammino verso modelli più efficaci per il riconoscimento automatico di scene di violenza richiederà un approccio multidisciplinare e iterativo, dove l'analisi delle sfide incontrate aiuterà a sviluppare soluzioni più robuste e adattabili.

Bibliografia

[1]Artificial Intelligence and Real Time Systems Laboratory, Dipartimento di Ingegneria dell'informazione, Università Politecnica delle Marche

[2]Github: https://github.com/airtlab/A-Dataset-for-Automatic-Violence-Detection-in-Videos

[3]U.S. Department of Justice, https://bjs.ojp.gov/document/cv22.pdf

[4]Simonyan, K., & Zisserman, A. Two-stream convolutional networks for action recognition in videos. In Advances in neural information processing systems (NeurIPS), 2014, (pp. 568-576).

[5]Wang, H., Kläser, A., Schmid, C., & Liu, C. L. Dense trajectories and motion boundary descriptors for action recognition. International Journal of Computer Vision (IJCV), 2013, 60-79.

[6]Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? A new model and the kinetics dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017 (pp. 6299-6308).

[7] Michael Nielsen, "Neural Networks and Deep Learning", Dec 2019.

[8]Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[9]Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

[10]Yeola, C. "Convolutional Neural Network (CNN) In Deep Learning." Python in Plain English, https://python.plainenglish.io/convolution-neural-network-cnn-in-deep-learning-77f5ab457166, 2022.

[11]Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," in IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, March 1994, doi: 10.1109/72.279181.

[12]Chung, Junyoung & Gulcehre, Caglar & Cho, KyungHyun & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling

[13]Abdelmoniem, Mai & Gasser, Safa & El-Mahallawy, Mohamed & Fakhr, Mohamed & Soliman, Abdel-Hamid. (2019). Enhanced NOMA System Using Adaptive Coding and Modulation Based on LSTM Neural Network Channel Estimation. Applied Sciences. 9. 3022. 10.3390/app9153022.

[14]C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308

[15]BERT: Pre-Training of Deep Bidirectional Transformers for language understanding (Devlin et al., NAACL 2019).

[16]Google Colab: https://colab.research.google.com/drive/1EEMIt7w58c9R2hxLpyktzGxtigxB3-ZQ

[17]Google Colab: https://colab.research.google.com/drive/15CIJU3p-dm99yqaLpAp8-CpeIEpI9-hi

[18]Documentazione di Keras: https://keras.io/api/models/model/#summary