Università degli Studi di Milano - Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di Laurea in Informatica

Riconoscimento di azioni da sequenze video digitali

Relatore: Prof. Schettini Raimondo

Co-relatore: Dott. Buzzelli Marco

Relazione della prova finale di:
Matteo Turla
Matricola 816235

Anno Accademico 2018-2019

Al Prof. Schettini e al Dott. Buzzelli, per la fiducia e l'impegno riposti in me e per la passione con cui affrontano il proprio lavoro.
Ai miei genitori, a mia sorella Martina e a mio fratello Andrea, per avermi sostenuto psicologicamente ed economicamente in questi tre anni e per avermi trasmesso la voglia di raggiungere questo traguardo.
A Gaia, per avermi supportato (e sopportato) moralmente e migliorato con fiducia e amore.
Ai miei amici, per aver reso questo percorso più leggero.
A Marco, per la disponibilità, la professionalità ed i preziosissimi insegnamenti, che hanno reso un piacere lavorare insieme a lui.
Ai miei nonni, che hanno sempre sognato e desiderato questo momento.
A Perla, che da quando è arrivata a casa non ha fatto altro che distrarmi.

Abstract

Partendo da un'attenta analisi della letteratura, sono state studiate e sviluppate tecniche di riconoscimento di azioni da sequenze video con l'obiettivo di monitorare le persone nella propria abitazione. Inoltre è stato dimostrato come lo *scheletro dinamico* possa essere una valida alternativa al costoso flusso ottico per modellare le caratteristiche temporali.
I risultati ottenuti mostrano come le soluzioni proposte possano formare una buona base di partenza per le ricerche future.

Indice

Capitolo 1

Introduzione

1.1 Riconoscimento di azioni

Ogni azione, sia semplice che complessa, è eseguita per uno scopo.
L'essere umano è in grado di riconoscere l'azione effettuata e lo scopo del soggetto che l'ha compiuta.
Usare, però, il lavoro umano per il monitoraggio delle azioni effettuate in una vastità di scenari reali è troppo dispendioso.
Può una macchina avere le stesse capacità dell'essere umano per compiere questo particolare task?

Il riconoscimento di azioni da sequenze video è l'obiettivo di un insieme di algoritmi che prendono in input un video e successivamente, dopo aver osservato una parte o l'intera esecuzione di una qualsiasi azione umana, producono un'ipotesi sull'azione svolta.
Il termine *azione umana*, studiato nella computer vision, spazia da semplici movimenti di arti ad azioni complesse che richiedono l'utilizzo di tutto il corpo. In media ogni azione viene rappresentata da video della durata di pochi secondi.
Non esiste una definizione formale di *azione umana*, ma ad ogni azione è associata una specifica tipologia tra le seguenti:
Individual actions: azioni che non presentano alcuna interazione con oggetti o altri individui; ad esempio camminare, sedersi o cadere.
Human interaction actions: azioni che presentano interazioni con altri soggetti; ad esempio stringere la mano, abbracciare o spingere qualcuno.
Object interaction actions: azioni che presentano interazioni con oggetti; ad esempio bere un bicchiere d'acqua o leggere un libro.

1.2 Applicazioni Real-World

Gli algoritmi di riconoscimento di azioni da sequenze video possono essere adoperati in moltissimi campi: dalla videosorveglianza[1] al monitoraggio delle persone nella propria abitazione[2].

Videosorveglianza
La sicurezza è un problema fondamentale ai giorni nostri ed è una tematica molto discussa. In alcuni ambienti posti sotto sorveglianza determinate azioni sono proibite.
Attraverso un sistema di sorveglianza basato sul riconoscimento di azioni da sequenze video[1] è possibile rilevare in tempo reale azioni pericolose o dannose e segnalare il soggetto che le compie.

Video retrival
L'enorme quantità di video caricati nel web ha reso la loro gestione molto difficoltosa.
Il riconoscimento di azioni, in questo caso, può essere utilizzato nella gestione ed estrazione dei video da un database[3]: attraverso l'analisi del contenuto del video è possibile associare a esso delle etichette di interesse, le quali sono utilizzate in fase di ricerca in modo tale da estrarre tutti i video appartenenti a una determinata categoria.

Videogame
L'interesse verso la realtà aumentata è in forte crescita e la nuova generazione di videogiochi è basata sul movimento del corpo umano. Il riconoscimento di azioni può essere adoperato per individuare i movimenti svolti dal giocatore.

1.3 Sfide e difficoltà

Il riconoscimento di azioni da video richiede di catturare il contesto dall'intera sequenza video. Può sembrare una naturale estensione del problema della classificazione di immagini, ma classificare ogni singolo frame e aggregare le singole predizioni sull'asse temporale non è una strategia ottimale in quanto questo modo di affrontare il problema non tiene conto delle informazioni temporali e delle caratteristiche di movimento del soggetto che compie l'azione.

Le maggiori difficoltà riscontrate sono:

- **Variabilità all'interno della stessa classe.**
  Ogni individuo può comportarsi in modo differente nello svolgere la medesima azione. L'azione *correre*, ad esempio, può essere svolta lentamente, velocemente e con diversi stili di movimento.
  La stessa azione può essere eseguita assumendo diverse posizioni: il soggetto può rispondere al telefono mentre cammina, da seduto oppure da sdraiato.
  Inoltre i video che riprendono la stessa azione possono avere diversi angoli di ripresa: dall'alto, dal basso, frontale o di lato, mostrando quindi variazioni evidenti della stessa azione.

- **Similarità fra classi diverse.**
  L'esecuzione di alcune azioni risulta molto simile, come ad esempio correre e camminare oppure bere e mangiare, in quanto hanno pattern di movimento quasi identici.

- **Mancanza di dataset ampi.**
  Lo sviluppo e il successo del deep learning è dovuto alla grande quantità di dati a

disposizione.
Invece, per quanto riguarda il riconoscimento di azioni da sequenze video, non sono ancora sufficienti i dati raccolti che catturano scenari reali. Questa mancanza di dati porta a significativi problemi nell'allenamento delle architetture utilizzate.

1.4 Obiettivi

L'obiettivo di questa ricerca è il monitoraggio di persone nelle loro abitazioni, al fine di individuare azioni di allerta e di tracciare il comportamento giornaliero del soggetto, attraverso l'utilizzo di algoritmi per il riconoscimento di azioni da sequenze video.
Tracciare il comportamento del soggetto monitorato rende possibile l'estrazione di statistiche utili, come il livello di idratazione del soggetto, o l'individuazione di possibili anomalie.

Capitolo 2

Analisi della letteratura

Per ottenere delle prestazioni elevate nel riconoscimento di azioni da sequenze video bisogna affrontare due problemi fondamentali: modellare sia le caratteristiche spazio-temporali del video che i pattern di movimento del soggetto che effettua l'azione.

2.1 Algoritmi Tradizionali

Gli algoritmi tradizionali sono algoritmi che, sebbene ancora oggi utilizzati, dominavano la scena prima dell'avvento del deep learning. Questi algoritmi prevedono tre diversi step:

1. Estrazione di feature locali di grandi dimensioni, sia sparse[4] che dense[5] [6], per descrivere regioni di video.

2. Combinazione delle feature estratte per formare una descrizione di lunghezza fissa del video. Una tecnica molto utilizzata per creare un vettore di feature di dimensione fissa è *Bag of visual words*[7].

3. Allenamento di un classificatore, come ad esempio *Support Vector Machine*[8], sulle feature combinate per la predizione della classe finale.

Tra gli algoritmi tradizionali quello che ottenne le migliori prestazioni fu *Improve Dense Trajectories*[9].

2.2 Deep learning

Con l'aumento dei dati a disposizione sono state studiate nuove tecniche di apprendimento automatico, le quali hanno ottenuto risultati nettamente superiori nel campo dell'elaborazione delle immagini e del linguaggio naturale rispetto agli algoritmi tradizionali.
Le reti neurali convoluzionali[10] sono una classe di modelli del deep learning che sostituiscono i 3 step di un algoritmo tradizionale attraverso un unico step corrispondente all'allenamento di una rete neurale basata su filtri convoluzionali. Questi modelli utilizzati riescono, in un unico step, a estrarre le feature, combinarle e classificarle.

2.2.1 Reti neurali convoluzionali

Le reti neurali convoluzionali, anche conosciute come CNNs, sono una tecnica di apprendimento automatico supervisionato e sono specializzate per processare dati che hanno una struttura a griglia, come ad esempio le immagini.
Le CNNs cercano di apprendere autonomamente le relazioni che sussistono tra i pixel, come ad esempio gli angoli o i bordi di un oggetto, e altre caratteristiche delle immagini usate in fase di allenamento con l'obiettivo di classificare immagini mai viste dalla rete.

L'obiettivo delle reti neurali è approssimare una *funzione obiettivo* f^*. Nel problema della classificazione, ad esempio, la funzione $y = f^*(x)$ mappa un input x a una categoria y. Una rete neurale definisce una funzione $y = f(x; \theta)$ e apprende i valori dei parametri θ con l'obiettivo di approssimare nel miglior modo la funzione obiettivo $y = f^*(x)$ riducendo il numero di classificazioni errate dando in input un insieme di dati di esempio.

Il nome *rete neurale convoluzionale* indica il fatto che la rete utilizza una particolare operazione matematica chiamata *convoluzione*.

2.2.2 Operazione di convoluzione

L'operazione di convoluzione costituisce l'unità base di tutti gli algoritmi della *computer vision* in quanto riesce a estrarre informazioni utili dalle immagini, come mostrato dalla Figura 2.1.

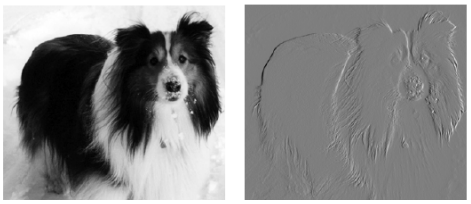

Figura 2.1: Risultato dell'estrazione dei bordi verticali attraverso l'operazione di convoluzione.

L'operazione di convoluzione sulle immagini modifica i valori di ogni pixel dell'immagine attraverso una funzione dipendente sia dal valore del pixel considerato che da quelli a esso vicini.
Eseguire l'operazione di convoluzione significa effettuare per ogni pixel dell'immagine una somma di prodotti tra l'intorno considerato del pixel dell'immagine f e un filtro w, anche chiamato kernel. L'intorno è stabilito dalla grandezza del filtro.
In generale l'operazione di convoluzione su un'immagine di dimensioni $M\ x\ N$ e un filtro di dimensioni $m\ x\ n$ dispari è definita come segue 2.1:

$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t) \tag{2.1}$$

dove x e y variano in modo tale che il centro del kernel visiti ogni pixel dell'immagine, generando una nuova immagine denominata *feature map*.

L'equazione 2.2 illustra l'operazione di convoluzione nel pixel (x, y) dell'immagine f e un kernel w di dimensioni 3 x 3. La risposta $g(x, y)$ è uguale alla somma dei prodotti tra i valori del kernel e l'intorno del pixel considerato.

$$\begin{aligned} g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + ... \\ +w(0,0)f(x,y) + ... + w(1,1)f(x+1,y+1) \end{aligned} \tag{2.2}$$

L'obiettivo delle *reti neurali convoluzionali* è apprendere autonomamente i valori dei parametri dei kernel in modo tale da estrarre le caratteristiche utili per classificare l'immagine analizzata.

2.2.3 Modelli per il riconoscimento di azioni da video

Nonostante le elevate prestazioni delle reti neurali convoluzionali nel riconoscimento di immagini, non è ancora chiaro il tipo di architettura da utilizzare per tutti quei problemi basati sulle sequenze video.
Alcune delle principali domande poste dai vari ricercatori sono:

1. Bisogna utilizzare filtri convoluzionali 2D o 3D?

2. L'input della rete deve essere un video RGB o deve anche includere il flusso ottico?

3. In caso di utilizzo di filtri convoluzionali 2D, come si propagano le informazioni temporali?

Le principali architetture presenti nella letteratura si possono suddividere in due categorie: architetture basate su filtri convoluzionali 2D [11] [12] o su filtri convoluzionali 3D [13] [14] [15].
Le architetture basate su kernel 2D estraggono, prima, le informazioni spaziali da singoli frame e, successivamente, aggregano i risultati per modellare le informazioni temporali del video.
Invece le architetture basate su kernel 3D modellano simultaneamente le caratteristiche spaziali e temporali lavorando su sequenze temporali di frame.

Per prima cosa sono state analizzate due reti che hanno posto le fondamenta di tutte le ricerche future, successivamente una rete convoluzionale 2D ricorrente, la quale rientra nelle architetture della prima categoria, e una rete convoluzionale 3D, la quale rientra nelle architetture della seconda categoria.

2.2.4 Single Stream Network

In *Large-scale Video Classification with Convolutional Neural Networks* [16] sono state analizzate varie tecniche di fusione delle informazioni spaziali, estratte dai singoli frame del video attraverso reti neurali convoluzionali 2D, per modellare la dimensione temporale.

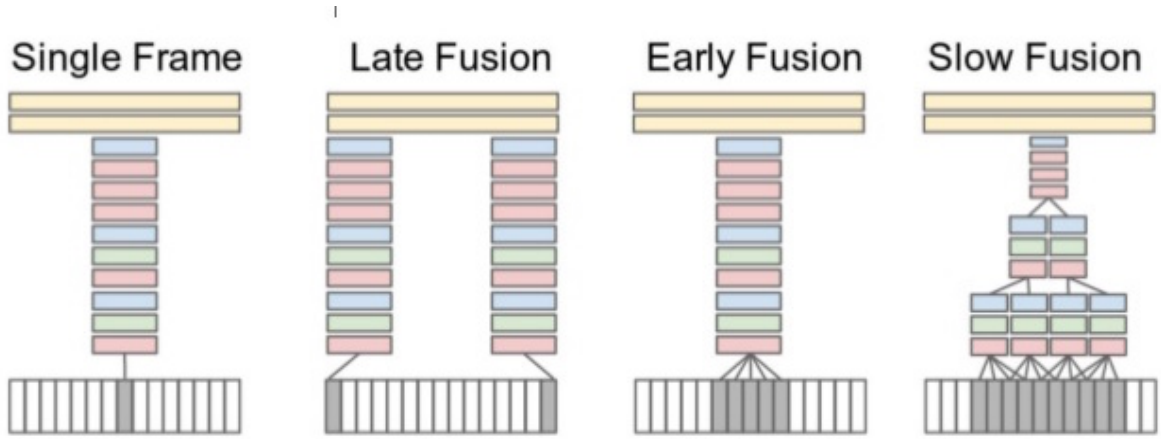

Figura 2.2: Metodi di fusione analizzati in *Large-scale Video Classification with Convolutional Neural Networks* [16].

Come si può vedere nella Figura 2.2, ogni tipologia di rete prende in input uno o più frame per poi fondere le informazioni estratte a diversi livelli.
Single-Frame. L'architettura single frame è una classica rete neurale convoluzionale che prende in input un'immagine statica. Viene utilizzata come benchmark per esaminare se le tre tecniche di fusione delle caratteristiche spaziali possano portare vantaggi nel modellare le informazioni temporali.
Early-Fusion. L'architettura early fusion combina le informazioni temporali e spaziali all'inizio del modello. Viene modificato il primo filtro convoluzionale dell'architettura single-frame in modo tale da prendere in input non più un'immagine statica, ma un insieme di T frame.
Late fusion. L'architettura late fusion utilizza due reti single stream che condividono gli stessi parametri. Prende in input coppie di frame distanti 15 frame l'uno dall'altro e ne combina le informazioni estratte soltanto alla fine.
Slow-fusion. L'architettura slow fusion è una combinazione tra il modello early fusion e late fusion. Questo modello prende in input 4 gruppi sovrapposti da 4 frame ciascuno e ne combina le informazioni a vari livelli di profondità.

Nonostante i numerosi esperimenti condotti da questa ricerca, i risultati sono ancora inferiori ai metodi tradizionali per due principali motivi: il dataset utilizzato per l'addestramento ha un numero di esempi non sufficiente per allenare una rete neurale e le architetture analizzate non riescono a catturare le informazioni di movimento dal video.

2.2.5 Two Stream Network

Dopo i fallimenti dell'architettura *Single stream network* dati dall'incapacità del modello di apprendere le caratteristiche di movimento, Simonyan et al. [17], invece di estrarre le informazioni temporali da un insieme di frame, ha deciso di modellarle attraverso il flusso ottico calcolato sull'intera sequenza video. L'architettura *Two Stream Network* si

basa, quindi, su due reti neurali convoluzionali separate: la prima si occupa di estrarre le informazioni spaziali, mentre la seconda quelle temporali, combinandole soltanto alla fine. La rete spaziale prende in input un singolo frame RGB, mentre quella temporale prende in input il flusso ottico estratto dall'intera sequenza video. Ogni rete è allenata separatamente e combinata soltanto alla fine.

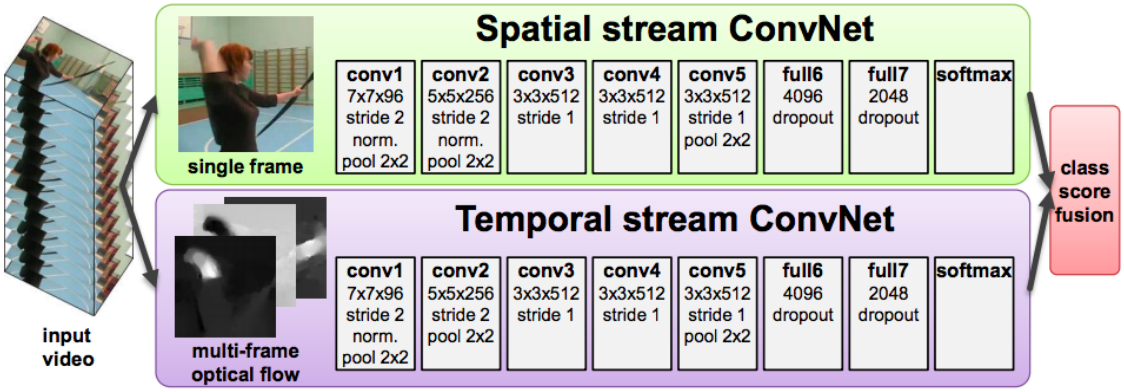

Figura 2.3: Architettura Two Stream Network.

Il metodo analizzato mostra che utilizzare il flusso ottico, in aggiunta ai frame RGB, porta ad un significativo aumento delle prestazioni.
Il flusso ottico risulta, però, costoso da calcolare e salvare.

2.2.6 Long-Term Recurrent Convolutional Networks

In *Long-Term Recurrent Convolutional Networks for Visual Recognition and Description*[12] è stato analizzato come combinare le informazioni estratte da ogni singolo frame del video in modo da modellarne le caratteristiche di movimento locali e globali.
L'architettura progettata modella il video come una sequenza ordinata di frame dai quali vengono estratte le feature attraverso una rete convoluzionale 2D. I risultati vengono successivamente composti per formare una sequenza temporale di feature che sarà l'input di una rete neurale ricorrente *long short term memory*[18].
Una rete neurale ricorrente[19] è in grado di modellare le informazioni temporali attraverso l'utilizzo di stati, i quali consentono alla rete di memorizzare le informazioni appena analizzate. L'output, quindi, non dipenderà solo dall'input attuale, ma anche dalle informazioni ricevute negli istanti di tempo passati. Questo permette alla rete di modellare un comportamento dinamico temporale.

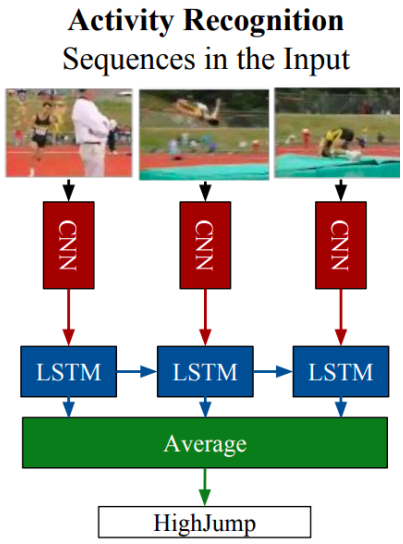

Figura 2.4: Architettura *Long-Term Recurrent Convolutional Networks*.

Nonostante la rete cerchi di modellare esplicitamente le informazioni temporali, non si riescono a catturare le informazioni temporali a lungo range o a modellare movimenti troppo dettagliati.
L'architettura risulta costosa da allenare in quanto utilizza due reti neurali, una indipendente dall'altra.

2.2.7 3D ConvNets

In *Learning Spatiotemporal Features With 3D Convolutional Networks* [20] è stato studiato come estendere le reti neurali convoluzionali 2D, le quali modellano solo le caratteristiche spaziali, a reti neurali convoluzionali 3D in modo tale da modellare simultaneamente le caratteristiche spaziali e temporali.
Un'immagine statica può essere vista come una matrice bi-dimensionale di pixel. Un video è una sequenza di immagini statiche, ovvero una sequenza di matrici bi-dimensionali, che formano, quindi, un volume di pixel.
L'operazione di convoluzione 2D è eseguita solo spazialmente, mentre l'operazione di convoluzione 3D è eseguita sia spazialmente che temporalmente. Come si vede dalla Figura 2.5, l'operazione di convoluzione 2D su una singola immagine o su una sequenza di frame, considerati come canali, produce un'immagine. Utilizzare, quindi, un'operazione di convoluzione 2D sui video causa la perdita delle informazioni temporali.
Solo l'operazione di convoluzione 3D produce come risultato un volume, preservando, così, le informazioni temporali e propagandole nel tempo.

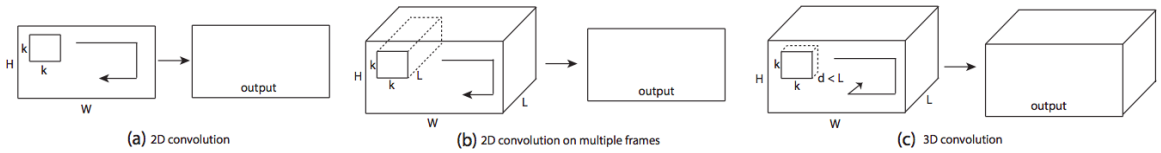

Figura 2.5: Diversi tipi di convoluzione.

In questo lavoro sono state analizzate le dimensioni che devono assumere i filtri convoluzionali per ottenere le migliori prestazioni.
L'architettura analizzata riesce a modellare sia le informazioni temporali che quelle spaziali senza il bisogno di utilizzare il flusso ottico.
Le prestazioni sono state relativamente basse a causa della mancanza di esempi reali attraverso cui allenare la rete.
Nonostante l'architettura sia considerata poco profonda, l'allenamento di questa rete è considerato molto dispendioso a causa dell'elevato numero di parametri.

Capitolo 3

Dataset

Il dataset, fornito dal *laboratorio di imaging e computer vision dell'università Milano-Bicocca*, si concentra su singole azioni umane rappresentate da video RGB.
Il dataset contiene un totale di 1193 video della lunghezza di pochi secondi, 813 video compongono il train set e 380 video compongono il test set.
I video riprendono, da varie prospettive, una singola persona svolgere un'azione in un contesto abitativo, risultando abbastanza puliti e non presentando evidenti problemi di rumore.
Le azioni svolte sono suddivise in 3 categorie: **Basic**, **Alerting**, **Daily life**. La tabella 3.1 mostra le azioni contenute in ogni categoria.

Action Classes		
Basic	Alerting	Daily Life
sitting down	sitting on flr.	drinking
standing up	standing up from flr.	eating
walking	lying down on flr.	reading
lying	falling	using phone
	touching body parts	wearing on/off
	vomiting/sneezing	using laptop
	coughing	exercising
	waving hands	

Tabella 3.1: Azioni presenti nel dataset suddivise per categoria.

La categoria **Basic** contiene azioni che portano il soggetto in uno stato diverso da quello precedente. Per stato si intende l'inattività del soggetto, come ad esempio *essere seduto* o *essere in piedi*. Le azioni contenute nella categoria *Basic* sono di tipo *Individual actions*.
La categoria **Alerting** rappresenta azioni che potrebbero portare il soggetto in una situazione pericolosa, come ad esempio *cadere* o *accasciarsi a terra*. Anche le azioni appartenenti a questa categoria sono di tipo *Individual actions*.
La categoria **Daily life** contiene quelle azioni che un soggetto svolge quotidianamente nella propria abitazione. Riconoscere questa tipologia di azioni è utile per tracciare un comportamento giornaliero della persona monitorata. Le azioni *daily life* sono di tipo *Object interaction actions*.

Le azioni presenti in *Basic* e *Alerting* sono azioni che richiedono maggiore ragionamento sul movimento effettuato, mentre le azioni in *Daily life* richiedono un'attenta analisi dell'oggetto con cui si sta interagendo. Azioni come *bere* e *mangiare*, ad esempio, presentano un pattern di movimento molto simile, ciò che le differenzia è l'oggetto utilizzato.

Statistiche train set					
	n. video	media frames	sdv frames	min 16 frames	min 32 frames
Basic	238	80.0	63.0	179	118
Alerting	325	67.0	24.0	325	307
Daily life	250	127.0	63.0	250	249

Tabella 3.2: Statistiche estratte dal train set per ogni categoria: numero di video, numero di frame medio per video, deviazione standard del numero di frame, numero di video con almeno 16 f. e numero di video con almeno 32 f.

Statistiche test set					
	n. video	media frames	sdv frames	min 16 frames	min 32 frames
Basic	120	95.0	69.0	92	75
Alerting	130	65.0	21.0	130	120
Daily life	130	151.0	85.0	130	130

Tabella 3.3: Statistiche estratte dal test set per ogni categoria: numero di video, numero di frame medio per video, deviazione standard del numero di frame, numero di video con almeno 16 f. e numero di video con almeno 32 f.

Come si può notare dalle tabelle 3.2 e 3.3, i video presentano una lunghezza molto variabile, sia all'interno della stessa classe che fra classi distinte, rendendo ardua la selezione degli iperparametri delle tecniche sviluppate.
Per ogni categoria il dataset è stato bilanciato ricopiando i video in modo tale che ogni classe abbia lo stesso numero di esempi.

3.1 Elaborazione del Dataset

Dal dataset di partenza sono stati costruiti tre diversi dataset: *Center video crop dataset*, *OpenPose skeleton dataset*, *Pose video crop dataset*. Tutte le operazioni di trasformazione descritte sono state effettuate sia sul train set che sul test set.
Center video crop dataset: Ogni video è stato ridimensionato in modo tale che la sua dimensione minore sia pari a 256 pixel, dopo di che è stato ritagliato al centro tramite un box di 224 x 224 pixel.
OpenPose skeleton dataset: contiene gli scheletri estratti dai singoli video originali attraverso l'utilizzo di OpenPose[21]. Lo scheletro è rappresentato da una matrice di grandezza 18 x 3 x T, dove T è il numero di frame.

Pose video crop dataset: utilizzando lo scheletro precedentemente estratto è stata calcolata la minima bounding box spazio-temporale quadrata. Ogni video è stato ritagliato con la corrispettiva bounding box e infine ridimensionato ad una dimensione di 224 x 224 pixel.

Per ogni categoria sono state analizzate e sviluppate due tecniche differenti per il riconoscimento di azioni da sequenze video e una combinazione tra le due: *Inflated 3D ConvNet*, *Spatio-Temporal Graph ConvNet*, *TwoStream ConvNet*

Capitolo 4

Inflated 3D ConvNet

Inflated 3D ConvNet[13] è una rete neurale convoluzionale 3D basata sull'espansione dei filtri di una rete neurale utilizzata per la classificazione di immagini: i filtri e i pooling kernel della rete inception-v1[22] sono espansi in 3D rendendo, così, possibile l'estrazione di caratteristiche spazio-temporali dai video e preservando il modello e i parametri perfezionati per il riconoscimento di immagini su ImageNet[23].

In questo capitolo è stata, per prima cosa, analizzata l'architettura e, successivamente, è stata implementata una possibile soluzione ponendo particolare attenzione all'obiettivo di monitorare le persone nella propria abitazione.

4.1 Descrizione del metodo

Negli anni sono state studiate numerose architetture per la classificazione di immagini attraverso numerosi tentativi ed errori, i quali hanno reso i modelli sempre più performanti. Invece di ripetere lo stesso processo anche per i modelli spazio-temporali, Carreira et al.[13] propone di convertire un modello di successo della classificazione di immagini in una rete convoluzionale 3D. Questo può essere eseguito aggiungendo un'ulteriore dimensione a tutti i filtri e i pooling kernel di un'architettura 2D.
I filtri, solitamente quadrati $N \times N$, vengono trasformati in filtri cubici $N \times N \times N$.

4.1.1 Inizializzare filtri 3D da filtri 2D

Per sfruttare tutti i vantaggi del modello di partenza è necessario preservare, oltre all'architettura, anche i pesi allenati su ImageNet[23].
Per fare ciò è stato osservato che un'immagine può essere convertita in un video attraverso la sua ripetizione fino a formare una sequenza temporale statica. Il modello 3D, allora, può essere implicitamente pre-allenato su ImageNet tramite la ripetizione dei valori dei pesi dei filtri 2D N volte sulla dimensione temporale e dividendoli per N. Facendo così la risposta ottenuta dai filtri convoluzionali sulle immagini è la medesima di quella ottenuta sui video.
Rimane da analizzare il modo in cui strutturare gli operatori di pooling sull'asse temporale e come impostare il valore di stride.
Se questi ultimi crescono troppo velocemente nel tempo rispetto allo spazio i bordi

delle regioni del video potrebbero confondersi e portare confusione nell'estrazione delle caratteristiche iniziali; se invece questi crescono troppo lentamente potrebbe essere difficile catturare la dinamicità della scena.
Carreira trova utile, dopo numerosi esperimenti, non utilizzare l'operazione di pooling sulla dimensione temporale per i primi due layer.

4.1.2 Two 3D stream

Nonostante il modello riesca a catturare le caratteristiche di movimento prendendo in input solamente il video RGB, l'utilizzo ulteriore del flusso ottico porta a prestazioni più elevate. Carreira decide, quindi, di utilizzare una configurazione a due rami identici: un ramo è allenato solamente su video RGB, mentre l'altro è allenato separatamente sul flusso ottico. Le predizioni delle due reti vengono poi mediate alla fine.

4.2 Esperimenti e Risultati

Per adattare il modello ai nostri obiettivi è stato utilizzato solo il ramo che prende in input il video RGB in quanto il flusso ottico risulta costoso da calcolare e salvare, inoltre non può essere utilizzato per approcci real-time.
Tutti gli esperimenti effettuati sono stati condotti utilizzando il modello pre-allenato su Kinetics[24]. Per poter confrontare i risultati, i vari esperimenti sono stati effettuati utilizzando gli stessi parametri durante la fase di allenamento.

Come si nota dalle statistiche del dataset originale, descritte nella Sezione 3.2, i video hanno un numero di frame molto variabile. Per ovviare a questo problema è stato deciso di utilizzare soltanto gli N frame centrali della sequenza, estratti come segue:
in fase di train sono stati selezionati N frame centrati in una posizione estratta da una distribuzione gaussiana, con media nel centro della sequenza e deviazione standard pari a 3.
In fase di test, invece, sono stati selezionati gli N frame centrali, dove il centro è esattamente il frame che divide a metà il video.

Come primo esperimento è stato allenato l'ultimo layer della rete variando il numero di frame che il modello prende in input, passando da 16 a 32. Non è stato possibile sperimentare la rete allenandola con 64 frame in quanto il dataset possiede esempi di lunghezza mediamente minore.
Il modello è stato allenato utilizzando *Center video crop dataset*, descritto nella Sezione 3.1.

16 vs 32 frames center accuracy		
	16 frames	32 frames
Alerting	0.6737	0.7803
Basic	0.8255	0.8666
Daily life	0.8424	0.9142

Tabella 4.1: Risultati ottenuti variando il numero di frame in input.

Come si può vedere dalla tabella 4.1, i risultati migliori sono stati ottenuti allenando la rete con video di 32 frame, poiché utilizzare 16 frame porta la rete a modellare solo una parte dell'azione, aumentando, così, la similarità tra le classi.
Tutti i successivi esperimenti sono stati effettuati utilizzando 32 frame, sia in fase di train che in fase di test.

Successivamente il modello è stato allenato cercando di simulare il suo reale utilizzo: in fase di inferenza il soggetto monitorato non è obbligato a svolgere l'azione al centro della telecamera o a una profondità fissa. Per rendere il modello indipendente da questi fattori è stato necessario individuare il soggetto che effettua l'azione. *Pose video crop dataset*, descritto nella Sezione 3.1, è stato creato tenendo conto delle considerazioni appena effettuate.
Il modello è stato allenato sull'ultimo layer utilizzando *Pose video crop dataset*, descritto nella Sezione 3.1, sia in fase di train che in fase di test.

Center crop vs Pose crop accuracy		
	Center crop	Pose crop
Alerting	0.7803	0.7614
Basic	0.8666	0.8416
Daily life	0.9142	0.8999

Tabella 4.2: Confronto delle prestazioni in base al dataset utilizzato - *Center video crop dataset* vs *Pose video crop dataset*.

Come si può notare dalla tabella 4.2, i risultati sono leggermente peggiorati. L'utilizzo, però, dell'architettura allenata su *Pose video crop dataset* è inevitabile in quanto simula con maggiore precisione i contesti per cui il sistema è stato progettato.
Infatti, in un contesto abitativo reale, la persona monitorata può muoversi in qualsiasi punto della stanza, uscire dalla visione della telecamera o essere in presenza di altre persone.
Risulta necessario, quindi, prima individuare dove avviene l'azione e poi classificarla.
Utilizzare il modello allenato su *Center video crop dataset*, descritto nella Sezione 3.1, in un contesto reale porta probabilmente a elevati errori di classificazione in quanto non tiene conto delle considerazioni appena effettuate.

Infine l'architettura è stata allenata su vari livelli della rete andando sempre più in profondità.
In questo esperimento è stato deciso di utilizzare *Pose video crop dataset* in quanto simula con maggiore precisione il reale utilizzo del sistema.

different layer fine-tuning accuracy			
	ultimo	ultimi 2	ultimi 3
Alerting	0.7614	0.8245	0.9000
Basic	0.8416	0.9541	0.9749
Daily life	0.8999	0.9142	0.9428

Tabella 4.3: Risultati ottenuti variando il numero di layer allenati.

Come si nota dalla tabella 4.3, i risultati migliori sono stati ottenuti allenando gli ultimi tre layer, ma non è stato possibile allenare la rete con ulteriore profondità per problemi di risorse.

4.3 Osservazioni

L'architettura analizzata, nonostante l'utilizzo del solo stream RGB, presenta ottimi risultati, riuscendo a modellare sia le caratteristiche spaziali che quelle temporali dei video, anche in presenza di azioni con caratteristiche simili.
Il modello risulta, però, costoso da allenare.
Tutti i modelli allenati presentano problemi di *overfitting* nonostante l'utilizzo di alti livelli di regolarizzazione. Per ovviare a questo problema è indispensabile l'aumento della cardinalità del dataset.

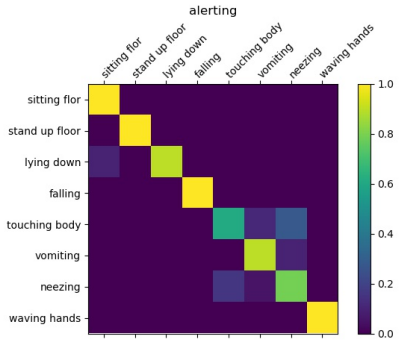

Figura 4.1: Matrice di confusione per la categoria Alerting, accuracy 0.9000

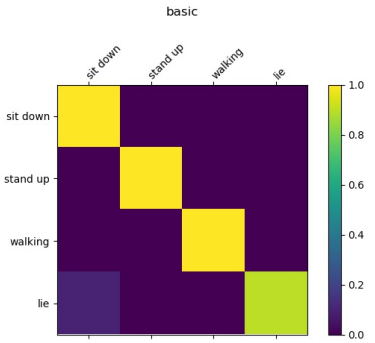

Figura 4.2: Matrice di confusione per la categoria Basic, accuracy 0.9747

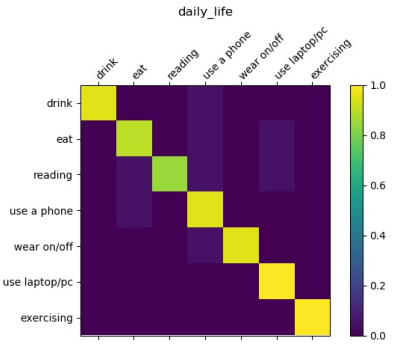

Figura 4.3: Matrice di confusione per la categoria Daily life, accuracy 0.9428

I risultati mostrati nelle figg. 4.1 4.2 4.1 sono stati ottenuti allenando gli ultimi tre layer del modello utilizzando *Pose video crop dataset* descritto nella Sezione 3.1 e passando in input 32 frame.

Capitolo 5

Spatio-Temporal Graph ConvNet

Spatio-Temporal Graph ConvNet[25], a differenza delle altre architetture basate sui video RGB e sui flussi ottici, cerca di apprendere automaticamente i pattern di movimento spaziali e temporali attraverso l'utilizzo dello scheletro dinamico estratto dai video RGB.

In questo capitolo è stata analizzata la costruzione dello *scheletro dinamico* partendo dai singoli video RGB e, successivamente, il modello è stato sperimentato variando diversi parametri.

5.1 Descrizione del metodo

5.1.1 Introduzione

Uno *scheletro* è un insieme di coordinate (x, y), le quali rappresentano le articolazioni dello scheletro umano, estratte da immagini statiche o video. Un algoritmo per l'estrazione dello scheletro, come ad esempio OpenPose[21], prende in input un'immagine o un video e restituisce un insieme di punti, uno per ogni articolazione individuata, rappresentati da coordinate x e y.
Uno *scheletro dinamico* è una sequenza temporale ordinata degli scheletri estratti dai singoli frame di un video.
L'obiettivo della seguente architettura è quello di apprendere automaticamente le caratteristiche di movimento dell'azione partendo dallo scheletro. Lo scheletro risulta robusto a cambi di illuminazione, scena e prospettiva ed è molto semplice da estrarre attraverso l'utilizzo di reti neurali, come ad esempio OpenPose[21]. Essendo lo scheletro un grafo non è possibile utilizzare reti neurali convoluzionali classiche in quanto, queste, prendono in input una griglia di pixel.
Le reti neurali su grafo sono una generalizzazione delle reti neurali convoluzionali e prendono in input un grafo con una struttura arbitraria.

5.1.2 Costruzione del grafo

Uno scheletro dinamico è una sequenza ordinata di scheletri ed è rappresentato da un insieme ordinato di coordinate 2D, le quali sono estratte da ogni frame del video. A partire dalla sequenza di scheletri, precedentemente estratta, viene costruito un grafo non

orientato G = (V,E).
Supponendo di utilizzare N articolazioni per rappresentare un singolo scheletro e T frame, l'insieme dei nodi V = $\{v_{ti}|t = 1, ..., T; i = 1, ..., N\}$ costituisce l'insieme di tutte le articolazioni estratte da ogni frame.
Le articolazioni del singolo frame sono connesse attraverso archi in accordo con la connettività dello scheletro umano, come mostrato dalla Figura 5.1.

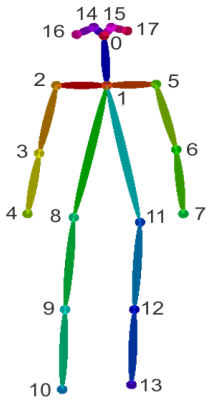

Figura 5.1: intra-body connections.

In seguito ogni articolazione del frame corrente è stata connessa alla corrispettiva articolazione del frame successivo, come mostrato nella Figura 5.2.

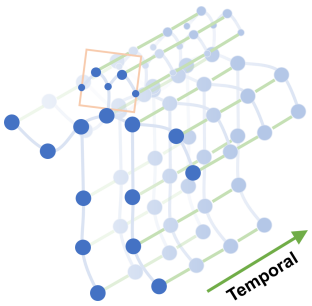

Figura 5.2: inter-frame connections.

Formalmente l'insieme degli archi E è composto da due sottoinsiemi: il primo è l'insieme delle connessioni *intra-body* $E_S = \{v_{ti}v_{tj}|(i, j) \in H\}$, dove H è l'insieme delle naturali connessioni delle articolazioni in uno scheletro umano. Il secondo sottoinsieme è quello delle connessioni *inter-frames* $E_F = \{v_{ti}v_{(t+1)i}\}$, le quali connettono le stesse articolazioni in frame consecutivi. Questo sottoinsieme rappresenta la traiettoria nel tempo dello scheletro.

5.2 Esperimenti e Risultati

Tutti gli esperimenti sono stati effettuati utilizzando il modello pre-allenato su Kinetics [24].
Per poter confrontare i risultati, i vari esperimenti sono stati effettuati utilizzando gli stessi parametri durante la fase di allenamento.
La rete è stata allenata utilizzando *OpenPose skeleton dataset*, come descritto nella Sezione

3.1, sulle sottosequenze centrali del video, come illustrato nella Sezione 4.2.
Come primo esperimento sono state confrontate le prestazioni in base alla profondità dell'input. La rete è stata allenata solo sull'ultimo layer.

16 vs 32 frames center accuracy		
	16 frames	32 frames
Alerting	0.7562	0.7915
Basic	0.8265	0.7963
Daily life	0.5857	0.5928

Tabella 5.1: Risultati ottenuti variando la profondità della sequenza temporale.

Come si nota dalla tabella 5.1, i risultati migliori sono stati ottenuti utilizzando 32 frame in quanto molte azioni presentano movimenti simili se considerate soltanto per pochi frame.

In seguito sono state analizzate le prestazioni ottenute variando il numero di layer allenati utilizzando in input 32 frame.

different layer fine-tuning accuracy					
	1	2	3	4	5
Alerting	0.7915	0.7864	0.8312	0.8801	0.8878
Basic	0.7963	0.9130	0.9291	0.9500	0.9666
Daily life	0.5929	0.6000	0.6428	0.6214	0.6214

Tabella 5.2: Risultati ottenuti variando il numero di layer allenati, i numeri identificativi delle colonne rappresentano quanti layer sono stati allenati partendo dall'ultimo.

La rete presenta prestazioni migliori quando viene allenata maggiormente in profondità, come mostrato dalla Figura 5.2. Essendo questa un'architettura poco profonda è stato deciso di non allenare ulteriormente la rete, così da non perdere i vantaggi del pre-allenamento su *Kinetics* [24].

5.3 Osservazioni

L'architettura analizzata prende in input sequenze temporali di scheletri, le quali riescono a modellare efficientemente il movimento del corpo del soggetto. Le reti neurali convoluzionali su grafi sono difficoltose da implementare e, a differenza delle CNNs, hanno una storia molto recente. Le prestazioni ottenute possono essere un ottimo punto di partenza per ricerche future.

Come si può notare dalle figg. 5.3 5.4 5.5, le azioni appartenenti alla categoria *daily life*, nelle quali esistono una forte componente di interazione con gli oggetti della scena, non riescono ad essere modellate. Questo è dovuto al fatto che lo scheletro rappresenta solo l'individuo presente nel video, tralasciando tutto il resto. Nelle azioni *bere* e *mangiare*, ad esempio, è di vitale importanza riconoscere l'oggetto con cui l'individuo interagisce in quanto tutte e due le azioni hanno un pattern di movimento molto simile.

Il modello riesce ad apprendere automaticamente i pattern di movimento delle azioni *single person* con prestazioni elevate. Per le azioni *human-interaction* e *object-interaction* sono necessarie ulteriori ricerche.

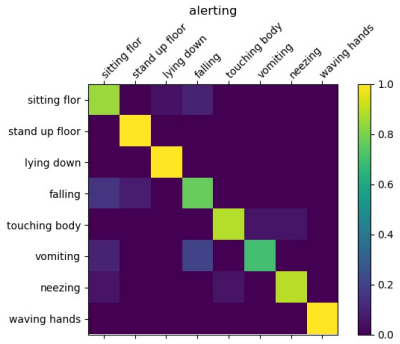

Figura 5.3: Matrice di confusione per la categoria Alerting, accuracy 0.8878

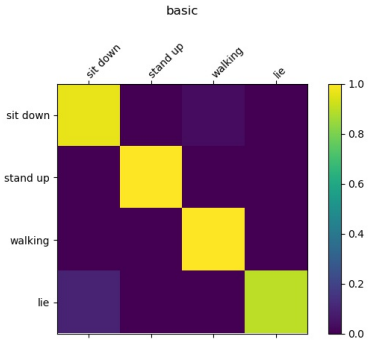

Figura 5.4: Matrice di confusione per la categoria Basic, accuracy 0.9666

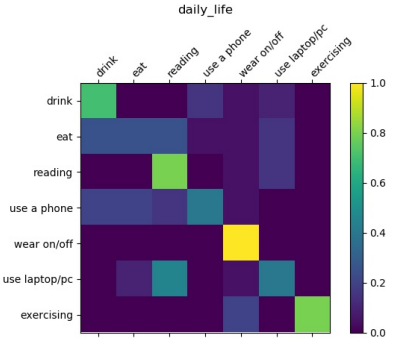

Figura 5.5: Matrice di confusione per la categoria Daily life, accuracy 0.6214

I risultati mostrati nelle figg. 5.3 5.4 5.5 sono stati ottenuti allenando gli ultimi cinque layer del modello utilizzando *OpenPose skeleton dataset* descritto nella Sezione 3.1 e passando in input una sequenza temporale di 32 frame.

Capitolo 6

TwoStream ConvNet

Le architetture che ottengono le migliori prestazioni nel riconoscimento di azioni da sequenze video sono quelle che adoperano nello stesso modello due rami distinti: un ramo prende in input il video RGB con l'obiettivo di estrarne le informazioni spaziali, mentre l'altro ramo prende in input il flusso ottico con l'obiettivo di estrarne le informazioni temporali.
Le predizioni dei due rami sono infine combinate.
Come dimostrato da Simonyan et al.[17], il flusso ottico migliora notevolmente le prestazioni delle architetture analizzate in quanto modella esplicitamente le caratteristiche temporali del video.
Il flusso ottico è, però, dispendioso da calcolare e salvare, inoltre non può essere utilizzato per approcci real-time.

Una possibile alternativa al flusso ottico può essere lo scheletro dinamico, calcolato come descritto nel capitolo 5. Lo scheletro è robusto a cambi di illuminazione, prospettiva e scena, inoltre è utile per individuare il soggetto spazialmente nel video, operazione descritta nella Sezione 3.1.
Nel capitolo 5 è stato dimostrato come lo scheletro dinamico riesca a modellare molto bene i pattern di movimento delle azioni *single person* confondendosi, però, con quelle azioni che presentano delle interazioni con vari oggetti.
L'obiettivo è individuare una valida alternativa al flusso ottico attraverso l'utilizzo dello scheletro dinamico, ma con un costo computazionale notevolmente minore.

In questo capitolo è stata analizzata e sviluppata una nuova architettura *TwoStream* basata su **Inflated 3D ConvNet** e **Spatio-Temporal Graph ConvNet**.

6.1 Esperimenti e risultati

Nel provare a combinare i due modelli sono state analizzate e implementate due possibili tecniche di fusione: *Media delle predizioni delle singole reti* e *Aggiunta di un nuovo layer per unire logicamente i due modelli*. Tutti gli esperimenti sono stati effettuati utilizzando i modelli, descritti e allenati nelle Sezioni 4.3 e 5.3, che hanno ottenuto le migliori prestazioni.

6.1.1 Media delle predizioni

In questo esperimento sono stati utilizzati i modelli precedentemente allenati, i quali vengono combinati mediando le predizioni effettuate separatamente dalle due reti.
Per calcolare le prestazioni sono stati utilizzati i test set di *OpenPose skeleton dataset* e di *Pose video crop dataset*. L'architettura sviluppata è stata denominata *TwoStream Mean*.

6.1.2 Aggiunta di un layer

In questo esperimento i due modelli sono stati fusi aggiungendo un layer finale, il quale prende in input la concatenazione delle feature estratte da *I3D* e *stgcn*. Per allenare l'ultimo layer sono stati utilizzati i modelli ottenuti come descritto nelle Sezioni 4.3 e 5.3. I dataset utilizzati sono: *Pose video crop dataset* e *OpenPose skeleton dataset*. L'architettura sviluppata è stata denominata *TwoStream ConvNet*.

6.2 Osservazioni

Come si può notare dalla tabella 6.1, la combinazione delle due architetture causa, in media, un piccolo aumento delle prestazioni rispetto all'impiego separato della rete *Inflated 3D ConvNet*, descritta nel Capitolo 4, e della rete *Spatio-temporal Graph ConvNet*, descritta nel Capitolo 5.
Aggiungere complessità alla rete crea problemi di overfitting, per questo motivo è stato deciso di utilizzare nel sistema progettato l'architettura TwoStream che combina le predizioni mediandole.

media vs nuovo layer				
	media	nuovo layer	I3D	stgcn
Alerting	0.9135	0.9157	0.9000	0.8878
Basic	0.9750	1.0000	0.9749	0.9666
Daily life	0.9500	0.8929	0.9428	0.6214

Tabella 6.1: Prestazioni ottenute mediando i risultati rispetto alle prestazioni ottenute aggiungendo un nuovo layer. Sono state riportate per confronto le prestazioni migliori ottenute da Inflated 3D ConvNet e quelle ottenute da Spatio-temporal Graph ConvNet.

Infine è stato effettuato un confronto tra *TwoStream Mean* e *I3D*: come si può vedere nella tabella 6.2, combinare i due modelli attraverso la media delle predizioni porta, oltre che a un piccolo aumento delle prestazioni, anche a un significativo aumento della confidenza media dovuto dal fatto che le due reti combinate *'sbagliano'* su azioni diverse, bilanciandosi l'una con l'altra.

I3D vs TwoStreamMean confidence				
	I3D accuracy	I3D confidence	TwoStreamMean accuracy	TwoStreamMean confidence
Alerting	0.9000	0.6073	0.9135	0.7338
Basic	0.9749	0.8982	0.9750	0.9416
Daily life	0.9428	0.5196	0.9500	0.6126

Tabella 6.2: Confronto delle confidenze medie ottenute da TwoStreamMean e I3D.

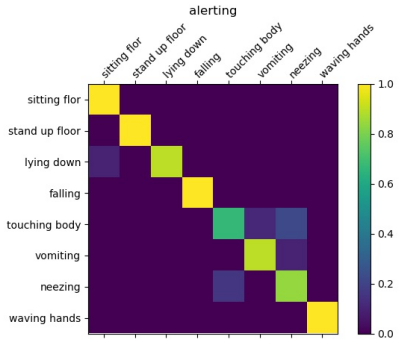

Figura 6.1: Matrice di confusione per la categoria *Alerting* ottenuta dalla rete **TwoStream Mean**, accuracy **0.9135**

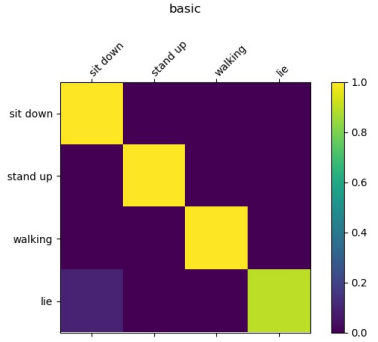

Figura 6.2: Matrice di confusione per la categoria *Basic* ottenuta dalla rete **TwoStream Mean**, accuracy **0.9750**

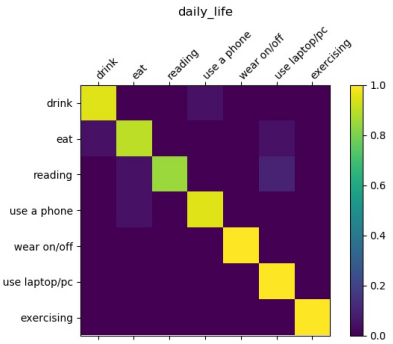

Figura 6.3: Matrice di confusione per la categoria *Daily life* ottenuta dalla rete **TwoStreamMean**, accuracy **0.9500**

Le migliori prestazioni, mostrate dalle figg. 6.1 6.2 6.3, sono state ottenute mediando le predizioni delle singole reti *Inflated 3D ConvNet* e *Spatio-temporal Graph ConvNet* allenate separatamente, come descritto rispettivamente nelle Sezioni 4.3 e 5.3.

6.3 Sistema progettato

Le ottime prestazioni ottenute da *TwoStream Mean* hanno reso possibile la progettazione e lo sviluppo di un sistema per il monitoraggio di una singola persona nella propria abitazione utilizzando le tecniche di riconoscimento di azioni da sequenze video appena analizzate.

La pipeline di esecuzione è la seguente:

1. **Acquisizione del video**
   Il sistema acquisisce, attraverso una telecamera, una sequenza video. Il video viene suddiviso in sottosequenze lunghe 32 frame con stride di 4 frame l'una dall'altra.

2. **Elaborazione della sottosequenza**
   Da ogni sottosequenza viene estratto lo scheletro del soggetto monitorato e, per mezzo di quest'ultimo, viene creata la minima bounding box spazio-temporale attraverso cui si ritaglia la sottosequenza in elaborazione. Attraverso lo scheletro estratto, inoltre, viene creato lo *scheletro dinamico* associato.

3. **Classificazione sottosequenza elaborata**
   Lo scheletro dinamico e la sottosequenza elaborata vengono processate da tre reti neurali *Two stream mean* identiche, una per ogni categoria di azione: *Alerting*, *Basic* e *Daily life*.
   Se la predizione migliore tra i tre output delle tre reti è maggiore di una certa soglia, fissata a 0.8, allora la sottosequenza in esame viene etichettata con l'azione predetta.

Capitolo 7

Conclusioni

Sono state analizzate e sviluppate varie tecniche di apprendimento automatico per il riconoscimento di azioni da sequenza video.
L'architettura **Inflated 3D ConvNet**, basata sul solo video RGB, è in grado di modellare le caratteristiche spazio-temporali con ottime prestazioni senza la necessità di utilizzare il flusso ottico; risulta però costosa da allenare ed è necessaria una maggiore quantità di esempi di addestramento.
L'architettura **Spatio-Temporal Graph ConvNet**, basata sullo scheletro dinamico estratto dal video RGB, modella efficientemente i pattern di movimento di azioni che non presentano interazioni con altri oggetti o persone.
Lo scheletro dato in input alla rete modella solamente il soggetto che effettua l'azione, tralasciando i possibili oggetti con cui una persona può interagire. Per questo motivo la rete risulta efficiente solo per il riconoscimento di azioni *single person*.
Tutti e due i modelli addestrati presentano notevoli problemi di *overfitting* nonostante l'utilizzo di forti livelli di regolarizzazione.
Infine sono state studiate e sviluppate due tecniche per la fusione dei modelli *I3D* e *st-gcn*: **TwoStream Mean** combina i modelli mediando le singole predizioni; **TwoStream ConvNet** combina i modelli classificando, attraverso un kernel 3D, le feature estratte dalle singole reti.
L'obiettivo della fusione delle due reti è ripercorrere i successi ottenuti dal flusso ottico superando, però, i problemi computazionali di quest'ultimo che non lo rendono adatto ad approcci *real-time*.
I risultati ottenuti mostrano come la rete **TwoStream Mean** sia in grado di modellare efficientemente qualsiasi tipo di azione, risolvendo i problemi di modellazione della rete *stgcn* e conseguendo un incremento delle prestazioni rispetto a quelle ottenute dalle singole reti.
Inoltre la confidenza media delle predizioni effettuate da **TwoStream Mean** è nettamente superiore rispetto a quella ottenuta da **I3D**, rendendo così possibile lo studio di una soglia per l'individuazione degli outlier.

Come risultato della ricerca è stato progettato e sviluppato un sistema per il monitoraggio delle persone nella propria abitazione, ottenendo ottimi risultati.

7.1 Ricerche future

Le ricerche future in questo campo sono molto ampie.
Il primo problema riscontrato è la mancanza di esempi che descrivano situazioni reali in un contesto abitativo. Un possibile sviluppo futuro potrebbe essere l'integrazione del dataset attuale con nuovi video.
Per quanto riguarda il sistema sviluppato può essere interessante aggiungere un tracking delle persone individuate in modo tale da consentire al sistema di lavorare con un numero variabile di persone nella scena e tracciare il comportamento di ogni singolo soggetto, come effettuato da Elmi et al.[2]
Per problemi di risorse non è stato possibile effettuare uno studio approfondito delle confidenze dei modelli allenati, ma un'ulteriore analisi potrebbe aumentare le prestazioni dei modelli attraverso l'individuazione degli outlier.

Bibliografia

[1] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[2] Gianluigi Ciocca, Alessio Elmi, Paolo Napoletano, and Raimondo Schettini. Activity monitoring from rgb input for indoor action recognition systems. In *2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, pages 1–4. IEEE, 2018.

[3] Mohsen Ramezani and Farzin Yaghmaee. A review on human action analysis in videos for retrieval applications. *Artificial Intelligence Review*, 46(4):485–514, 2016.

[4] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action recognition by dense trajectories. 2011.

[5] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. VS-PETS Beijing, China, 2005.

[6] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005.

[7] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[8] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[9] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

[11] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 4041–4049, 2015.

[12] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[13] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[14] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in neural information processing systems*, pages 3468–3476, 2016.

[15] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[16] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[17] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf`.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[19] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[20] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[21] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017.

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[23] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[24] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[25] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.