



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



CORSO DI LAUREA TRIENNALE IN INGEGNERIA INFORMATICA

Sviluppo di un sistema di riconoscimento di azioni tramite sensori inerziali

LAUREANDO

Giovanni Cinel

Matricola 2000147

RELATORE

Prof. Stefano Ghidoni

Università degli studi di Padova

CORRELATORE

Matteo Terreran, PhD

Università degli studi di Padova

ANNO ACCADEMICO
2022/2023

*Alla mia famiglia
che mi ha sempre sostenuto.*

Sommario

L'Human Action Recognition (HAR) è un campo di ricerca in rapida evoluzione che sta trasformando profondamente il nostro modo di interagire con il mondo digitale e le macchine. Questa tesi si concentra sullo sviluppo di un sistema di riconoscimento di azioni umane basato sull'utilizzo di sensori inerziali, ovvero sensori dotati di accelerometri e giroscopi. Il riconoscimento di azioni tramite sensori inerziali è una sfida complessa a causa della varietà di movimenti umani e delle azioni eseguibili. Questo studio esamina gli approcci principali di elaborazione dei segnali e di machine learning utilizzati per estrarre le caratteristiche dai dati sensoriali e addestrare le reti neurali. Attraverso un'attenta progettazione sperimentale e la raccolta di dati per vari tipi di azioni e soggetti, è stato sviluppato un algoritmo di riconoscimento delle azioni che ha raggiunto un'accuratezza del 91.58%. L'accuratezza del modello di riconoscimento ottenuta è ottima e garantisce un riconoscimento affidabile delle azioni, tuttavia, esistono ancora sfide e possibili sviluppi futuri nella ricerca del riconoscimento di azioni, in particolare per la classificazione di movimenti molto simili tra loro.

Indice

Elenco delle figure	xi
1 Introduzione	1
1.1 Applicazioni del riconoscimento di gesture e human action recognition in diversi settori	2
1.1.1 HAR nella robotica collaborativa	3
1.1.2 HAR nella sicurezza	4
1.1.3 HAR e salute	5
1.1.4 HAR e intrattenimento	6
2 Stato dell'arte	9
2.1 Tipologie di sensori utilizzati nel riconoscimento delle azioni umane	10
2.1.1 Telecamere RGB	10
2.1.2 Sensori infrarossi passivi (PIR)	11
2.1.3 Sensori inerziali	12
3 Metodi tradizionali e apprendimento automatico nel riconoscimento di gesture e human action recognition	15
3.1 Classificazione	15
3.2 Metodi tradizionali nel riconoscimento di gesture e human action recognition	17
3.2.1 Elaborazione dell'immagine	17
3.2.2 Tracciabilità del movimento	18
3.3 Approcci basati sull'apprendimento automatico nel riconoscimento di gesture e azioni umane	19
3.3.1 Reti neurali convoluzionali	20
3.3.2 Reti neurali ricorrenti	22
3.3.3 Reti neurali densamente connesse	23

INDICE

4 Setup sperimentale ed acquisizione del dataset	27
4.1 Tecnologia e set-up impiegati per la creazione del dataset	27
4.2 Dataset	30
4.3 Acquisizione dati	32
5 Sviluppo della rete neurale	37
5.1 Conversione dei dati	37
5.2 Allenamento della rete	38
5.2.1 Risultati ottenuti dalle reti neurali allenate	43
5.3 Risultati ottenuti	45
6 Conclusione e sfide aperte nel campo del riconoscimento di gesture e human action	47
6.1 Conclusione	47
6.2 Sfide aperte e prospettive future nel riconoscimento di gesture e human action recognition	48
Bibliografia	51
Ringraziamenti	55
A Appendice A: Codice in linguaggio C++ sviluppato per eseguire la conversione dei dati in formato utile per Edge Impulse	57

Elenco delle figure

1.1	HAR nella robotica collaborativa.	4
1.2	HAR nella sicurezza.	5
1.3	HAR in campo medico.	6
1.4	HAR nel campo dell'intrattenimento.	7
3.1	Architettura di una Rete Neurale Convoluzionale.	20
3.2	Max e mean pooling.	21
3.3	Architettura di una Rete Neurale Ricorrente.	23
3.4	Architettura di una Rete Densamente Connessa.	25
4.1	Mtw Awinda Dev. kit.	28
4.2	Xsens MTw Awinda.	28
4.3	Awinda Station.	29
4.4	Chiavetta USB Awinda.	29
4.5	Set cinturini in velcro.	29
4.6	Software XSens MVN.	29
4.7	Azioni del dataset raccolto presso lo IAS-Lab.	31
4.8	Soggetti partecipanti alla creazione del dataset.	32
4.9	Configurazione per le acquisizioni del terzo soggetto.	32
4.10	Sensori nella regione anteriore.	33
4.11	Sensori nella regione posteriore.	33
4.12	Posizione del soggetto descritta dai giunti.	34
5.1	Performance ottenute con il set di validazione.	41
5.2	Elementi del training set classificati dalla rete neurale.	42
5.3	Risultati ottenuti con il test set.	45

1

Introduzione

È inevitabile che la tecnologia di riconoscimento delle gesture e dell'active recognition continui a progredire e ad evolversi, aprendo nuovi orizzonti nell'interazione tra uomo e macchina e consentendo un'esperienza più naturale e fluida per l'utente.

- Thad Starner

Il riconoscimento di azioni, anche detto Human Action Recognition (HAR), è un ambito di ricerca emergente e in continua evoluzione che sta profondamente trasformando il modo in cui interagiamo con il mondo digitale e le macchine. L'obiettivo principale dell'HAR è quello di interpretare i movimenti del corpo umano e riconoscere le diverse azioni eseguite mediante l'utilizzo di sensori, algoritmi di machine learning e tecniche avanzate di elaborazione dei segnali. La rapida crescita e i progressi in questo settore hanno suscitato grande interesse nella ricerca sulla computer vision e sull'intelligenza artificiale, spianando la strada a un'ampia gamma di applicazioni innovative, che spaziano dalla realtà virtuale e aumentata, alla medicina riabilitativa, fino alla robotica collaborativa.

Una delle principali sfide dell'HAR nel campo della robotica collaborativa consiste nel progettare e sviluppare algoritmi di riconoscimento accurati e affidabili, che possano funzionare in diversi contesti e adattarsi alle diverse caratteristiche dei soggetti coinvolti. In particolare, questi algoritmi devono essere in grado di riconoscere le azioni utili per agevolare la collaborazione tra uomo e robot, garantendo al contempo l'esecuzione di attività in modo efficiente e sicuro, in spazi condivisi e dinamici.

1.1. APPLICAZIONI DEL RICONOSCIMENTO DI GESTURE E HUMAN ACTION RECOGNITION IN DIVERSI SETTORI

L'HAR è una tecnologia che può essere utilizzata anche per monitorare la salute [18] e l'attività fisica delle persone [26], per migliorare le performance sportive [14] o per migliorare il controllo di macchinari industriali [6].

Al fine di sviluppare un modello in grado di riconoscere in modo efficace le azioni umane, è fondamentale l'acquisizione di dati tramite sensori, i quali vengono raccolti in dataset, che verranno utilizzati attraverso algoritmi di machine learning per l'allenamento e la validazione del modello. Tra le diverse tecniche utilizzate per il riconoscimento di azioni, si annoverano l'impiego di sensori inerziali, sensori dotati di accelerometri e giroscopi, che forniscono la posizione e l'orientamento dei sensori stessi. Inoltre, la rilevazione del movimento può essere ottenuta anche mediante l'utilizzo di sensori di profondità, telecamere ed altre tipologie di sensori.

Tuttavia, il riconoscimento di azioni basato su sensori inerziali rappresenta ancora una sfida significativa, a causa della complessità dei movimenti del corpo umano e della vasta gamma di azioni che è possibile eseguire. Di conseguenza, è necessario impiegare tecniche avanzate di elaborazione dei segnali e di machine learning che permettano di riconoscere in modo accurato e affidabile le azioni eseguite dall'utente.

Questa tesi si propone di esplorare le tecniche di riconoscimento delle azioni umane nel contesto sportivo, concentrandosi specificamente sull'analisi delle azioni tramite l'utilizzo di sensori inerziali, utilizzando dati ottenuti mediante acquisizioni su tre soggetti.

1.1 APPLICAZIONI DEL RICONOSCIMENTO DI GESTURE E HUMAN ACTION RECOGNITION IN DIVERSI SETTORI

Il riconoscimento di gesture e HAR ha molte applicazioni in diversi settori, tra cui la robotica collaborativa, la sicurezza, la salute e il fitness, l'interazione uomo-computer e l'intrattenimento.

Il processo di riconoscimento può riguardare diversi tipi di movimenti che possono essere classificati gerarchicamente in base alla loro complessità come segue [5]:

- Azioni: consistono in semplici attività atomiche come saltare, correre o lanciare. Di solito, soggetti diversi eseguono la stessa azione in modo simile.

- Gesture: movimenti del corpo finalizzati a una comunicazione non verbale intenzionale o involontaria, di solito eseguiti con le mani o con la parte superiore del corpo. La loro esecuzione può essere universale o influenzata dall'ambiente sociale e culturale, oltre che dal giudizio individuale. Ad esempio, un gesto con significato di "saluto" può essere eseguito stringendo la mano, alzando il braccio o facendo un cenno con la testa. Il riconoscimento di gesture può essere utilizzato per controllare dispositivi, come i sistemi di domotica [13], o per comunicare con altre persone attraverso dispositivi tecnologici, come gli smartphone.

1.1.1 HAR NELLA ROBOTICA COLLABORATIVA

Il campo della robotica collaborativa sta diventando sempre più importante in molteplici ambiti, dai servizi di assistenza ai compiti di produzione, consentendo un'interazione tra uomo e robot sempre più naturale e intuitiva. Una parte fondamentale di questo campo è rappresentata dal riconoscimento di attività umane, che permette di identificare le azioni dell'utente e di fornirgli un supporto adeguato da parte del robot in modo sempre più efficiente ed efficace, consentendo ai robot di interagire in modo sempre più naturale e sicuro con gli esseri umani, aprendo dunque nuove opportunità per l'automazione di processi produttivi e di servizio [19]. Ad esempio, il robot potrebbe svolgere una funzione di assistente durante un assemblaggio, riconoscendo le azioni in corso svolte dall'operatore umano per anticiparne le future necessità, quali il passaggio del prossimo pezzo da assemblare o di un particolare strumento (e.g., cacciavite).

Il principale obiettivo dell'HAR è quello di dotare i robot di una maggiore consapevolezza dell'ambiente circostante, degli oggetti e delle persone che li circondano, al fine di garantire una interazione fluida e sicura tra robot e umani. Grazie all'utilizzo di tecniche avanzate di intelligenza artificiale e di sensori sofisticati, l'HAR rappresenta una soluzione innovativa per migliorare la qualità della vita umana, aumentare l'efficienza produttiva e creare nuove opportunità di lavoro.

1.1. APPLICAZIONI DEL RICONOSCIMENTO DI GESTURE E HUMAN ACTION RECOGNITION IN DIVERSI SETTORI

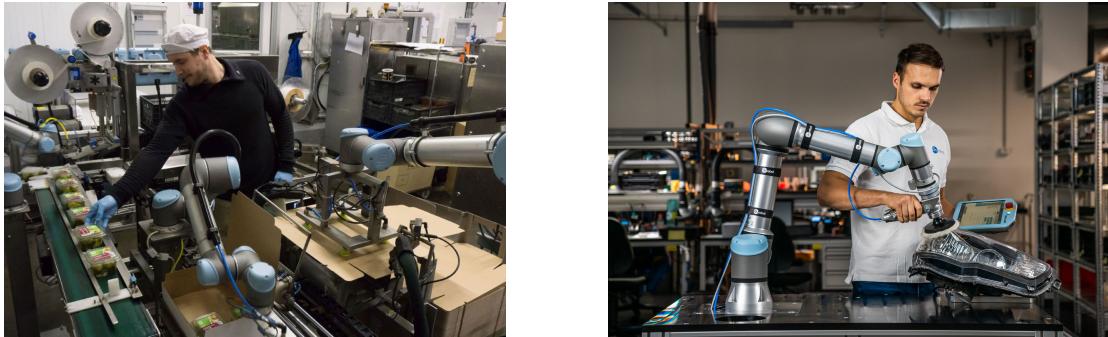


Figura 1.1: HAR nella robotica collaborativa.

1.1.2 HAR NELLA SICUREZZA

Nell’ambito della sicurezza, il riconoscimento delle gesture e l’HAR costituiscono una tecnologia innovativa e altamente promettente per il monitoraggio e la prevenzione di situazioni di pericolo in luoghi pubblici. In particolare, queste tecniche possono essere utilizzate per identificare comportamenti anomali o minacciosi, o per sorvegliare i confini e gli accessi in aree ad alto rischio.

Queste tecnologie possono essere utilizzate per creare un sistema di sorveglianza altamente efficace per la prevenzione e la gestione di situazioni di pericolo in luoghi pubblici [28]. Ad esempio, un sistema di riconoscimento delle gesture può essere utilizzato per identificare comportamenti sospetti o minacciosi, come ad esempio coprire o nascondere le braccia durante un taccheggio [10], mentre un sistema di HAR può essere utilizzato per monitorare l’attività fisica delle persone e rilevare eventuali comportamenti anomali e illeciti, come ad esempio azioni che implicano violenza [1] o l’abbandono di oggetti sospetti [25] potenzialmente esplosivi. Inoltre, queste tecnologie possono essere utilizzate anche per sorvegliare i confini e gli accessi in aree ad alto rischio come aeroporti, stazioni ferroviarie e porti marittimi. Ad esempio, un sistema di sorveglianza basato sull’HAR può essere utilizzato per monitorare l’attività degli operatori di sicurezza o del personale di supporto.

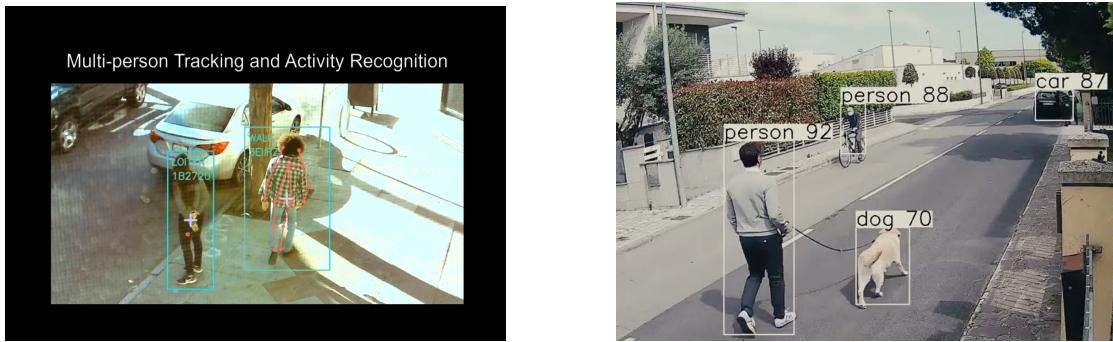


Figura 1.2: HAR nella sicurezza.

1.1.3 HAR e SALUTE

L'utilizzo dell'HAR nel settore della salute e del fitness è sempre più diffuso grazie alle sue numerose applicazioni. Ad esempio questi sistemi si sono dimostrati molto utili per fornire un'efficiente assistenza domiciliare agli anziani e sistemi di localizzazione in ambienti chiusi [17]. Inoltre, l'HAR può essere utilizzato per creare programmi di allenamento personalizzati, in base alle esigenze e alle capacità individuali del paziente.

Un altro vantaggio dell'HAR riguarda la sua capacità di fornire dati sull'attività fisica quotidiana [17]. Questo è particolarmente utile per i pazienti che necessitano di monitorare la loro attività fisica per motivi di salute.

Inoltre, l'HAR può essere utilizzato per supportare la riabilitazione fisica [11], fornendo feedback in tempo reale sulle prestazioni del paziente. Ciò aiuta a migliorare la tecnica e accelerare il processo di recupero.

Tuttavia, nonostante i numerosi vantaggi, ci sono ancora molti limiti da affrontare. Ad esempio, la precisione dei dati raccolti dipende dalla qualità dei sensori indossabili o dei dispositivi mobili utilizzati. Inoltre, l'interpretazione dei dati richiede competenze specialistiche e una buona conoscenza dell'anatomia umana.

In conclusione, l'HAR rappresenta una tecnologia innovativa e promettente nel campo della salute e del fitness. Grazie alle sue numerose applicazioni, può aiutare a migliorare la salute e il benessere dei pazienti, ma sono necessarie ulteriori ricerche per sviluppare algoritmi di apprendimento automatico sempre più precisi e affidabili.

1.1. APPLICAZIONI DEL RICONOSCIMENTO DI GESTURE E HUMAN ACTION RECOGNITION IN DIVERSI SETTORI



Figura 1.3: HAR in campo medico.

1.1.4 HAR e INTRATTENIMENTO

Negli ultimi anni, le tecnologie di Human Action Recognition e il riconoscimento delle gesture hanno guadagnato crescente interesse nell'industria dell'intrattenimento e dell'IT. Queste innovazioni tecnologiche hanno permesso di sviluppare sistemi di interazione uomo-macchina più efficaci, offrendo esperienze d'intrattenimento più immersive ed emozionanti. Le console videoludiche moderne utilizzano sempre più tecnologie di riconoscimento delle azioni umane e delle gesture per offrire una gameplay più avvincente enfatizzando l'interazione del giocatore con il gioco stesso. Un esempio emblematico è rappresentato dall'introduzione di dispositivi come il Kinect di Microsoft, utilizzato dalla console Xbox, e il PlayStation Move di Sony, che sfruttano rispettivamente tecnologie ad infrarossi e di movimento per tracciare le azioni del giocatore e tradurle in comandi all'interno del gioco.

La realtà virtuale (VR) e aumentata (AR) sono due tecnologie emergenti nel settore dei videogiochi, che si avvalgono di HAR e del riconoscimento delle gesture per offrire esperienze di gioco sempre più realistiche e coinvolgenti. Attraverso l'utilizzo di appositi visori e controller di movimento, dotati di sensori inerziali, i giocatori possono interagire con gli ambienti virtuali in maniera intuitiva e naturale, utilizzando le proprie mani e il corpo per manipolare oggetti, spostarsi e interagire con gli elementi di gioco. Attraverso l'analisi dei dati di accelerazione e rotazione forniti dai sensori, un sistema di HAR può riconoscere i movimenti della testa e delle mani dell'utente nel contesto della realtà virtuale. Queste informazioni consentono di traslare e ruotare l'ambiente virtuale in modo coerente con i movimenti dell'utente, garantendo un'esperienza più realistica e coinvolgente. Ad esempio, se l'utente inclina la testa verso sinistra, il sistema di

navigazione VR può modificare la visuale di conseguenza, fornendo l’illusione di un’interazione naturale e intuitiva con l’ambiente virtuale [8].

Le tecnologie di HAR e riconoscimento delle gesture hanno anche trovato applicazioni nell’ambito di concerti, spettacoli teatrali e installazioni artistiche interattive. Ad esempio, le performance di danza e musica possono essere rese più dinamiche e innovative grazie all’uso di sensori in grado di rilevare i movimenti dei performer e convertire questi dati in effetti visivi, sonori o di animazione in tempo reale.

Inoltre, le istituzioni culturali come musei e gallerie d’arte si avvalgono delle tecnologie di HAR per creare esperienze interattive e coinvolgenti per i visitatori. Spesso utilizzando gli occhiali VR o AR, questi ambienti culturali offrono la possibilità di esplorare le collezioni esistenti o di interagire con mostre virtuali e di apprendere in modo innovativo. L’uso di HAR in questo contesto permette di creare esperienze personalizzate e coinvolgenti, adattate agli interessi dei visitatori [27]. Questo ha gettato le basi per lo sviluppo di un turismo culturale intelligente [21], che offre agli utenti del turismo un senso di realtà e, allo stesso tempo, riduce notevolmente le risorse consumate per costruire una varietà di scene nella realtà. Inoltre, i dati raccolti attraverso HAR possono essere utilizzati per migliorare l’esperienza dei visitatori, fornendo feedback utili sulle aree che richiedono miglioramenti e consentendo ai curatori di comprendere meglio le preferenze del pubblico. In questo modo, l’HAR sta diventando una tecnologia sempre più importante per l’industria culturale.



Figura 1.4: HAR nel campo dell’intrattenimento.

2

Stato dell'arte

Il riconoscimento di gesture e azioni è una tecnologia che consente di identificare ed interpretare i movimenti e le azioni eseguite dall'uomo. Ad esempio, un'applicazione pratica dell'HAR riguarda il monitoraggio dell'attività fisica, grazie ad un sistema indossabile, come ad esempio un braccialetto fitness dotato di sensori inerziali, in grado di rilevare accelerazioni lungo gli assi x, y e z. Sfruttando un algoritmo di apprendimento automatico opportunamente addestrato, tale sistema può identificare e distinguere attività quali squat, push-up, curl e NAE(Not An Exercise) [20].

Il primo passo nel riconoscimento di gesture e HAR è l'acquisizione dei dati attraverso sensori, come le telecamere o i sensori di movimento, che registrano le azioni e le traducono in segnali elettrici o digitali, che possono essere interpretati da algoritmi di elaborazione. Gli algoritmi di elaborazione sono programmi che elaborano i dati, tramite algoritmi matematici e di intelligenza artificiale, per identificare ed interpretare le gesture e le azioni umane.

Le tecniche utilizzate nel riconoscimento di gesture e HAR sono in continua evoluzione, grazie all'avvento di algoritmi di deep learning e all'utilizzo di sensori sempre più avanzati. Le applicazioni del riconoscimento di gesture e HAR sono numerose e in costante espansione, ma ci sono anche molte sfide aperte data la complessità delle gesture umane, la sensibilità al rumore dei sensori e la necessità di una grande quantità di dati per l'apprendimento automatico.

Con il continuo sviluppo della tecnologia, il riconoscimento di gesture e HAR avrà un ruolo sempre più importante nella nostra vita quotidiana e nella nostra interazione con le macchine.

2.1 TIPOLOGIE DI SENSORI UTILIZZATI NEL RICONOSCIMENTO DELLE AZIONI UMANE

Nel campo dell'human action recognition e del riconoscimento di gesture vengono utilizzati diversi tipi di sensori per raccogliere dati necessari alla creazione di dataset da utilizzare per lo sviluppo di sistemi di riconoscimento di azioni. Alcuni esempi di sensori utilizzati per la raccolta dati includono:

- **Telecamere RGB:** utilizzate per registrare video di persone che eseguono azioni o gesture. I dati raccolti vengono impiegati per addestrare algoritmi di riconoscimento delle azioni e delle gesture registrate [4].
- **Sensori ad infrarossi passivi (PIR):** questi sensori rilevano le variazioni di temperatura causate dal movimento delle persone. Sono spesso utilizzati in sistemi di sicurezza domestica come sensori di movimento. L'utilizzo di sensori PIR nell'HAR ha inoltre suscitato un enorme interesse per le sue ampie applicazioni nell'Internet delle cose (IoT), nelle case intelligenti e nella sanità [7].
- **Sensori inerziali:** sono costituiti da un insieme di sensori che rilevano accelerazioni lineari e angolari, variazioni della velocità angolare e il campo magnetico terrestre per orientare il corpo nello spazio. Hanno alcune limitazioni come la complessità di calibrazione, la sensibilità alle vibrazioni e il drift dei giroscopi [12].

Ci sono anche sensori più avanzati che possono essere utilizzati per rilevare il movimento e la posizione delle mani e delle dita, come i **guanti con sensori di movimento integrati**, utilizzati per raccogliere dati per il riconoscimento delle gesture delle mani.

La scelta della tipologia migliore di sensore da utilizzare per una raccolta di dati richiede quindi di effettuare una valutazione accurata dei requisiti funzionali del sistema in cui il sensore sarà impiegato, al fine di soddisfare le richieste dell'applicazione in cui si vuole utilizzarlo. La scelta del tipo di sensore da utilizzare dipende dalla specifica esigenza dell'applicazione e dalla disponibilità delle risorse. Pertanto, occorre considerare i vantaggi e le limitazioni dei singoli dispositivi, ponendo in essere un'attenta analisi tecnica per selezionare il sensore più idoneo al fine di garantire la massima affidabilità del sistema.

2.1.1 TELECAMERE RGB

Le telecamere RGB rappresentano un tipo comune di sensori utilizzati per il riconoscimento dell'azione umana. In particolare, queste telecamere forniscono un'immagine dettagliata delle persone che eseguono le azioni, rendendole

un'ottima scelta per una vasta gamma di applicazioni, come la sorveglianza del territorio, la navigazione autonoma dei robot, la sicurezza industriale e molto altro ancora.

Nonostante permettano di ottenere una elevata qualità di immagine, le telecamere RGB hanno alcuni svantaggi che vanno presi in considerazione. Ad esempio l'utilizzo di questo tipo di sensori può richiedere uno spazio considerevole di archiviazione, per il salvataggio dei dati raccolti.

In sintesi, le telecamere RGB rappresentano una soluzione comune per il riconoscimento dell'azione umana, ma richiedono una attenta ed efficiente gestione dei dati raccolti.

Il riconoscimento dell'azione umana vede l'utilizzo di telecamere RGB come sensori in un'ampia gamma di settori, tra cui sorveglianza [3], assistenza ai pazienti, ad esempio [2] propone un nuovo sistema di assistenza per i pazienti malati di Alzheimer valutato sul dataset fornito dal progetto DemCare¹.

2.1.2 SENSORI INFRAROSSI PASSIVI (PIR)

Un'alternativa sono i sensori a infrarossi passivi (PIR), meno costosi e meno onerosi in termini di spazio di archiviazione rispetto alle telecamere RGB.

Tuttavia questa tipologia di sensori trova delle limitazioni a causa della vulnerabilità a interferenze elettriche ed elettromagnetiche e alla possibilità di essere influenzati dalla temperatura ambiente e da altri fattori ambientali, come le correnti d'aria e la luce solare diretta, che possono causare rilevazioni imprecise.

Nonostante ciò, i sensori PIR sono spesso utilizzati in applicazioni di sicurezza domestica [24], dove vengono impiegati per rilevare la presenza di movimenti all'interno di una stanza o di un'abitazione.

Nella progettazione di un sistema basato su sensori PIR, è importante considerare anche le limitazioni di questi sensori e le situazioni in cui potrebbero non fornire informazioni sufficienti. Ad esempio, i sensori PIR potrebbero non rilevare azioni di persone in determinate condizioni ambientali, potrebbe dunque essere necessario integrare i sensori PIR con altri dispositivi, come telecamere RGB, per ottenere informazioni più precise sull'azione.

In conclusione, i sensori a infrarossi passivi rappresentano un'alternativa alle telecamere RGB per il riconoscimento dell'azione umana, ma necessitano

¹<https://demcare.eu/datasets/>

2.1. TIPOLOGIE DI SENSORI UTILIZZATI NEL RICONOSCIMENTO DELLE AZIONI UMANE

di eventuali integrazioni con altri dispositivi per garantire la completezza delle informazioni raccolte.

2.1.3 SENSORI INERZIALI

I sensori di movimento inerziali, o IMU (Inertial Measurement Unit), sono costituiti da un insieme di sensori in grado di rilevare accelerazioni lineari e angolari e variazioni della velocità angolare. Gli IMU utilizzano le leggi di Newton per calcolare la posizione, la velocità e l'accelerazione di un corpo in movimento. L'utilizzo di IMU è comune in applicazioni come la navigazione inerziale, la realtà virtuale [9] e l'analisi del movimento umano [29].

Gli IMU sono composti dai seguenti sensori:

- **Accelerometro:** rileva le accelerazioni lineari utilizzando la forza di gravità come riferimento. Viene utilizzato per rilevare il movimento del corpo in uno spazio tridimensionale.
- **Giroscopio:** rileva le variazioni della velocità angolare utilizzando il principio di Coriolis. Viene utilizzato per rilevare la rotazione del corpo attorno ai tre assi.
- **Magnetometro:** rileva il campo magnetico terrestre e viene utilizzato come riferimento per l'orientamento del corpo nello spazio.

Gli IMU possono essere utilizzati per acquisire segnali come l'accelerazione e velocità lungo gli assi x,y e z, consentendo di classificare le azioni umane come camminare, correre, salire le scale e altre attività, in base all'andamento dei segnali acquisiti. Tuttavia, gli IMU hanno alcune limitazioni come la complessità di calibrazione, la sensibilità alle vibrazioni e il drift dei giroscopi. Per ovviare a queste limitazioni, gli IMU sono spesso integrati con altri sensori, come i sensori ottici di rilevamento della posizione e gli accelerometri ad alta precisione.

Gli IMU sono ampiamente utilizzati per le applicazioni che richiedono un'alta precisione nella rilevazione delle azioni umane come la navigazione in realtà virtuale e l'analisi del movimento degli atleti [23]. Tuttavia il loro costo può risultare proibitivo ed inoltre richiedono una calibrazione accurata per fornire risultati affidabili.

GUANTI CON SENSORI DI MOVIMENTO

Infine, i guanti con sensori di movimento integrati rappresentano una soluzione basata su IMU, specializzata per la rilevazione delle gesture delle mani [16].

Rappresentano quindi una soluzione vantaggiosa se l'applicazione di interesse riguarda solo le mani, tuttavia non sono molte le applicazioni che si limitano alle mani, richiedendo un livello di dettaglio così preciso.

In definitiva, la scelta del tipo di sensore dipende dalle specifiche esigenze dell'applicazione e dalla disponibilità delle risorse. Per scegliere il sensore più adatto alla situazione è quindi necessario valutare attentamente i requisiti funzionali del sistema in cui il sensore sarà impiegato, al fine di soddisfare le richieste delle applicazioni in cui si vuole utilizzarlo, ponderando vantaggi e limitazioni dei singoli dispositivi. I sensori che si presentano maggiormente adatti per il sistema di riconoscimento il cui sviluppo è riportato in questa tesi sono i sensori IMU e le telecamere RGB, tra questi si è scelto l'utilizzo di sensori inerziali che presentano un costo di archiviazione dei dati nettamente inferiore senza rinunciare alla qualità dei dati acquisiti.

3

Metodi tradizionali e apprendimento automatico nel riconoscimento di gesture e human action recognition

In questo capitolo viene presentata una panoramica delle tecniche usate comunemente in letteratura per il riconoscimento di azioni, verrà poi approfondito nel capitolo 5 il metodo utilizzato per lo sviluppo del sistema di riconoscimento presentato in questa tesi, ovvero una rete neurale densamente connessa.

3.1 CLASSIFICAZIONE

La classificazione nel machine learning è un processo che mira a assegnare un'etichetta, selezionandola tra un insieme predefinito di etichette, o una categoria a un dato oggetto o istanza in base alle sue caratteristiche o attributi. Questa tecnica è ampiamente utilizzata per la risoluzione di problemi di natura decisionale o di previsione, in cui è necessario assegnare un'etichetta a nuovi dati in base a pattern identificati nei dati di addestramento.

Per effettuare la classificazione, viene utilizzato un algoritmo di apprendimento supervisionato che richiede un training set e un test set, si parla di apprendimento supervisionato quando l'insieme delle etichette è noto. Il training set è un insieme di dati precedentemente etichettati, che viene utilizzato per addestrare il modello di classificazione. Ogni elemento del training set è composto da un insieme di attributi o caratteristiche e dall'etichetta corrispondente. L'obiettivo del modello di classificazione è apprendere i pattern o le

3.1. CLASSIFICAZIONE

regole presenti negli elementi del training set in modo da poter generalizzare e classificare correttamente nuovi dati.

Dopo aver addestrato il modello, viene utilizzato il test set per valutare le prestazioni del modello stesso. Il test set è un insieme di dati separato dal training set, che contiene anche le etichette corrispondenti. Questo set di dati viene utilizzato per testare il modello e valutare la sua capacità di generalizzazione. Il modello applica le regole apprese durante l'addestramento al test set e assegna le etichette predette a ciascuna istanza. Le etichette predette vengono quindi confrontate con le etichette effettive nel test set per valutare l'accuratezza del modello di classificazione.

L'obiettivo principale della classificazione nel machine learning è quello di creare un modello che possa generalizzare e assegnare correttamente etichette a nuovi dati ancora non visti. Per raggiungere questo obiettivo, il modello di classificazione deve essere addestrato su un training set rappresentativo e deve essere valutato attentamente utilizzando un test set indipendente. Questo processo consente di identificare eventuali problemi di sovraccarico (overfitting) o sottodattamento (underfitting) del modello e di apportare le correzioni necessarie per migliorare le prestazioni del classificatore.

Nel machine learning, esistono diversi algoritmi di classificazione che possono essere selezionati in base alle caratteristiche specifiche del dataset e del problema da affrontare. L'obiettivo principale è creare un modello predittivo accurato che possa generalizzare bene su nuovi dati, classificandoli correttamente in base alle categorie desiderate.

Alcuni dei principali algoritmi di classificazione includono:

- **Alberi di decisione:** modello di classificazione simile a un diagramma di flusso, dove le foglie rappresentano le categorie o classi e le diramazioni indicano le condizioni delle caratteristiche.
- **K-Nearest Neighbors (KNN):** è un algoritmo di classificazione basato sulle distanze. Un oggetto da classificare viene comparato con i suoi vicini più vicini e viene assegnato alla categoria o classe maggiormente presente.
- **Support Vector Machine (SVM):** è un algoritmo di classificazione che cerca di trovare il miglior separatore tra le varie categorie o classi. L'obiettivo è quello di identificare un iperpiano che separi i dati delle diverse classi.
- **Reti neurali:** un insieme di algoritmi che simulano il funzionamento del cervello umano, sono composte da unità chiamate neuroni artificiali o nodi, che sono collegati tra loro in strati.

Nell'esperienza sperimentale riguardante lo sviluppo di un sistema di riconoscimento di azioni umane, presentata in questa tesi, sono state utilizzate le reti neurali come algoritmo di classificazione.

3.2 METODI TRADIZIONALI NEL RICONOSCIMENTO DI GESTURE E HUMAN ACTION RECOGNITION

I metodi tradizionali nel riconoscimento di gesture e HAR utilizzano tecniche di elaborazione dell'immagine e di tracciamento del movimento. Queste tecniche sono basate sulla segmentazione dei movimenti, sulla definizione di parametri geometrici delle gesture e sulla classificazione statistica delle azioni.

3.2.1 ELABORAZIONE DELL'IMMAGINE

L'elaborazione dell'immagine è un processo complesso che comprende una serie di operazioni matematiche e algoritmiche volte ad analizzare e manipolare le informazioni visive contenute in un'immagine digitale. L'obiettivo dell'elaborazione dell'immagine è quello di ottenere informazioni utili a partire dall'immagine stessa, come ad esempio riconoscere oggetti, individuare forme o colori, o ancora identificare le gesture in un'immagine. Il processo di elaborazione dell'immagine può essere diviso in tre fasi principali: pre-processing, processing e post-processing.

La fase di pre-processing è la prima fase dell'elaborazione dell'immagine e consiste nel preparare l'immagine per la fase successiva di processing. Questa fase prevede la rimozione del rumore dall'immagine, la correzione dell'illuminazione, la riduzione della dimensione dell'immagine e la conversione del formato dell'immagine. Il pre-processing può anche prevedere la segmentazione dell'immagine, ovvero la suddivisione dell'immagine in parti separate per analizzarle individualmente.

La fase di processing è la fase centrale dell'elaborazione dell'immagine e consiste nell'applicare un'ampia gamma di tecniche di analisi dell'immagine per estrarre le informazioni di interesse dall'immagine. Questa fase prevede l'applicazione di filtri, l'analisi delle feature dell'immagine, la segmentazione dell'immagine, la classificazione e la correlazione tra oggetti.

3.2. METODI TRADIZIONALI NEL RICONOSCIMENTO DI GESTURE E HUMAN ACTION RECOGNITION

La fase di post-processing è l'ultima fase dell'elaborazione dell'immagine e prevede l'applicazione di tecniche di modifica dell'immagine in base alle informazioni ottenute nella fase di processing. Questa fase prevede la ricostruzione dell'immagine, la rimozione di eventuali errori residui e la rappresentazione dell'immagine in un formato finalizzato.

L'elaborazione dell'immagine si avvale di una vasta gamma di tecniche e algoritmi, tra cui la trasformata di Fourier, la segmentazione, il filtraggio e la classificazione. Tali tecniche possono essere applicate su diverse tipologie di immagini, come immagini a colori, immagini in scala di grigi o immagini binarie.

3.2.2 TRACCIABILITÀ DEL MOVIMENTO

La tracciabilità del movimento è un altro importante aspetto dei metodi tradizionali. La tracciabilità del movimento è il processo di tracciare l'evoluzione del movimento nel tempo, andando a determinare la posizione e l'orientamento di un oggetto in movimento utilizzando dati provenienti da sensori o da immagini acquisite da telecamere. Consentendo quindi di rilevare gli spostamenti e le variazioni di posizione delle regioni di interesse. Questa tecnica è particolarmente utile per riconoscere gesture che coinvolgono il movimento delle mani e delle braccia.

Il processo di tracciamento del movimento prevede la definizione di un modello matematico che descriva il movimento dell'oggetto in questione. Questo modello può essere basato su una serie di equazioni che rappresentano la dinamica dell'oggetto, oppure può essere ottenuto mediante tecniche di machine learning che apprendono il modello a partire dai dati di tracciamento. Una volta definito il modello di movimento, il processo di tracciamento può essere eseguito utilizzando diversi algoritmi.

Tuttavia, il tracciamento del movimento può essere una sfida complessa a causa di vari fattori, come la presenza di rumore nei dati del sensore, la presenza di ombre o riflessi, e la complessità del movimento dell'oggetto. Pertanto, sono necessarie tecniche avanzate di elaborazione dei segnali e di machine learning per garantire un tracciamento preciso e affidabile del movimento.

3.3 APPROCCI BASATI SULL'APPRENDIMENTO AUTOMATICO NEL RICONOSCIMENTO DI GESTURE E AZIONI UMANE

Gli approcci basati sull'apprendimento automatico nel riconoscimento di gesture ed azioni hanno recentemente raggiunto risultati molto promettenti grazie all'utilizzo di tecniche di deep learning. Gli algoritmi di deep learning sono in grado di apprendere automaticamente le caratteristiche delle immagini e dei segnali, senza la necessità di definirle in modo esplicito. Ciò ha permesso di superare le limitazioni dei metodi tradizionali e di migliorare l'efficacia del riconoscimento.

In particolare, le reti neurali convoluzionali (CNN) sono state ampiamente utilizzate per il riconoscimento di gesture e HAR. Questi algoritmi sono in grado di apprendere automaticamente le caratteristiche spaziali delle immagini e dei video, consentendo di riconoscere gesti complessi anche in presenza di rumore. Le reti neurali ricorrenti (RNN), invece, sono state utilizzate per riconoscere azioni che si evolvono nel tempo, come camminare o parlare.

Tra le diverse architetture di reti neurali, una delle più comuni è la rete neurale densamente connessa, anche nota come Multi-Layer Perceptron (MLP), rete utilizzata nello sviluppo del sistema di riconoscimento di azioni presentato al capitolo 5.

Inoltre, la combinazione di diverse modalità di input, come l'immagine, il suono e i sensori di movimento, ha permesso di migliorare ulteriormente l'efficacia del riconoscimento di gesture e HAR. Ad esempio, gli algoritmi di fusion sono stati utilizzati per combinare le informazioni derivanti da una telecamera e un radar a onde millimetriche, andando a migliorare le prestazioni nella classificazione di azioni acquisite in ambienti poco luminosi, situazione nella quale il solo uso di telecamere porta a ottenere prestazioni di basso livello [15].

In generale, gli approcci basati sull'apprendimento automatico nel riconoscimento di gesture e HAR sono ancora oggetto di ricerca attiva e si prevede che l'utilizzo di tecniche sempre più avanzate, come le reti neurali convoluzionali 3D e le reti generative, consentirà di raggiungere risultati sempre migliori in termini di precisione e di flessibilità nell'identificazione di gesture e azioni umane.

3.3.1 RETI NEURALI CONVOLUZIONALI

Le reti neurali convoluzionali sono un tipo di rete neurale artificiale che si basa sull'utilizzo di convoluzioni per l'estrazione delle caratteristiche. Questo tipo di rete è stato sviluppato per risolvere problemi di classificazione di immagini, ma oggi viene utilizzato anche in altri campi, come il riconoscimento del parlato e la traduzione automatica.

Le CNN sono costituite da diversi strati, visibili nella figura 3.1, ognuno dei quali ha una funzione specifica.

Il primo strato è il livello di input, che riceve l'immagine da classificare.

Il secondo strato è il livello di convoluzione, che applica vari filtri all'immagine per estrarre le caratteristiche principali.

Il terzo strato è il livello di pooling, che riduce la dimensione spaziale delle caratteristiche estratte, per fornire informazioni sull'input a varie risoluzioni e mantiene solo le caratteristiche più importanti.

Il quarto strato è il livello completamente connesso, che utilizza le informazioni estratte dai livelli precedenti per effettuare la classificazione dell'input nelle classi di interesse scelte per il problema in esame.

Questi strati sono organizzati in modo concatenato, l'output di uno strato viene utilizzato come input del successivo, al fine di estrarre gerarchicamente le caratteristiche delle immagini.

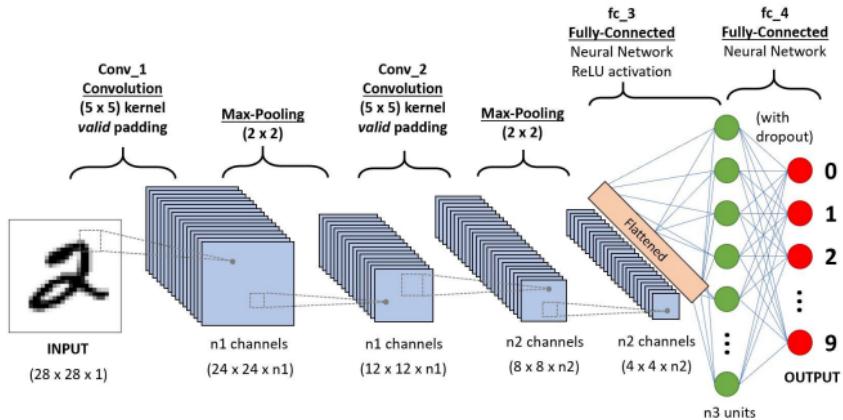


Figura 3.1: Architettura di una Rete Neurale Convoluzionale.

Nelle CNN, i livelli o strati sono definiti e organizzati in modo concatenato, ma spesso vengono ripetuti più volte per aumentare la complessità e la precisione del modello. Ad esempio, possono essere utilizzati più strati di convoluzione

in successione per estrarre caratteristiche sempre più complesse dell’immagine, oppure gli strati di pooling possono essere ripetuti per ridurre ulteriormente le dimensioni dell’immagine e aumentare la velocità di calcolo. La scelta del numero di livelli e della loro organizzazione dipende dal tipo di problema di classificazione dell’immagine e dai dati a disposizione.

La convoluzione è l’operazione principale delle CNN. Essa consiste nell’applicare un filtro all’immagine per estrarre le caratteristiche principali. Il filtro è costituito da una matrice di valori, chiamata kernel, che viene spostata sull’immagine in modo da coprire tutte le possibili posizioni. In ogni posizione, il prodotto tra il kernel e l’area dell’immagine coperta dal kernel viene sommato per ottenere un singolo valore di output. Questo processo viene ripetuto per tutte le posizioni dell’immagine, generando una nuova immagine la quale rappresenta le caratteristiche principali.

Il pooling è un’operazione che riduce la dimensione dell’immagine mantenendo solo le caratteristiche più importanti. Esistono diversi tipi di pooling, come il max pooling e il mean pooling. Nel max pooling, viene selezionato il valore massimo di ogni area dell’immagine, mentre nel mean pooling viene calcolata la media dei valori di ogni area dell’immagine.

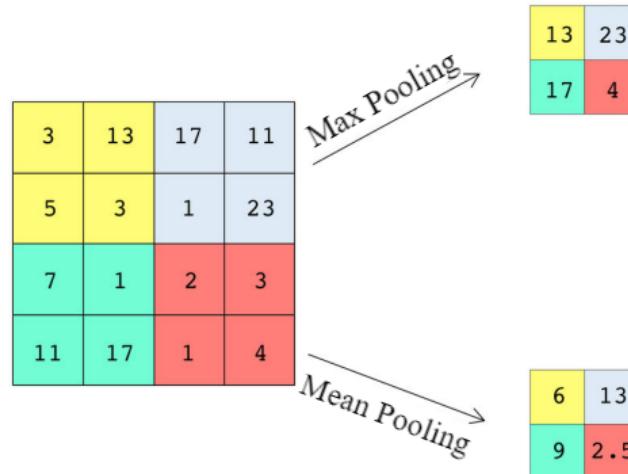


Figura 3.2: Max e mean pooling.

Le CNN utilizzano l’apprendimento supervisionato per imparare a classificare le immagini. Durante la fase di addestramento, la rete viene alimentata con un insieme di immagini di input e le rispettive etichette associate. La re-

3.3. APPROCCI BASATI SULL'APPRENDIMENTO AUTOMATICO NEL RICONOSCIMENTO DI GESTURE E AZIONI UMANE

te cerca quindi di minimizzare l'errore tra le etichette previste e quelle reali, aggiornando i pesi dei vari strati attraverso la retropropagazione dell'errore.

Le CNN hanno diversi vantaggi rispetto ad altre tecniche di classificazione delle immagini. In primo luogo, sono in grado di estrarre automaticamente le caratteristiche principali delle immagini, senza la necessità di specificare manualmente le caratteristiche da cercare, necessario invece con le tecniche tradizionali. In secondo luogo, sono in grado di gestire immagini di dimensioni diverse, grazie all'utilizzo del pooling. Infine, sono in grado di gestire anche immagini con rumore o distorsioni, grazie alla capacità di generalizzazione della rete.

In sintesi, le CNN sono un tipo di rete neurale artificiale che si basa sull'utilizzo di convoluzioni per l'estrazione delle caratteristiche principali delle immagini. Grazie alla loro capacità di generalizzazione e alla loro flessibilità, le CNN sono utilizzate in diversi campi per risolvere problemi di classificazione e di intelligenza artificiale.

3.3.2 RETI NEURALI RICORRENTI

Le reti neurali ricorrenti sono un tipo di rete neurale artificiale che si basa sull'utilizzo di feedback per elaborare sequenze di dati. Questo tipo di rete è stato sviluppato per risolvere problemi di elaborazione del linguaggio naturale, ma oggi viene utilizzato anche in altri campi, come la previsione finanziaria e la modellizzazione del clima.

Le RNN sono costituite da diversi strati, ognuno dei quali ha una funzione specifica.

Il primo strato è il livello di input, che riceve la sequenza di dati da elaborare.

Il secondo strato è il livello ricorrente, che utilizza una memoria interna per mantenere informazioni sulle sequenze precedenti.

Il terzo strato è il livello completamente connesso, che utilizza le informazioni estratte dai livelli precedenti per effettuare la previsione o la classificazione.

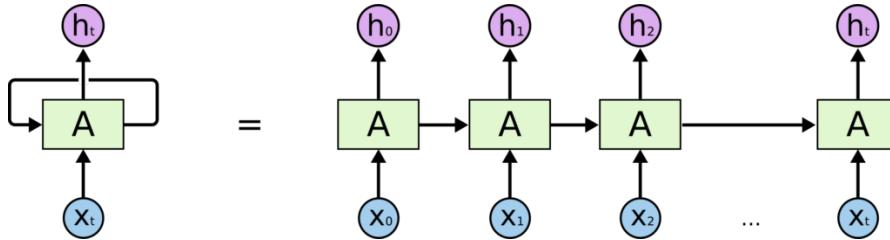


Figura 3.3: Architettura di una Rete Neurale Ricorrente.

Il feedback, operazione principale delle RNN, consiste nell'utilizzare la memoria interna del livello ricorrente per elaborare le sequenze di dati. In ogni passaggio, il livello ricorrente riceve in input il dato corrente e la memoria interna relativa al passaggio precedente. Utilizzando queste informazioni, il livello ricorrente aggiorna la memoria interna e produce un output che viene utilizzato dal livello completamente connesso.

Le RNN utilizzano l'apprendimento supervisionato per imparare a elaborare le sequenze di dati. Durante la fase di addestramento, la rete viene alimentata con un insieme di sequenze di input e le rispettive etichette associate o previsioni. La rete cerca quindi di minimizzare l'errore tra le etichette previste e quelle reali, aggiornando i pesi dei vari strati attraverso la retropropagazione dell'errore.

Le RNN hanno diversi vantaggi rispetto ad altre tecniche di elaborazione delle sequenze di dati. In primo luogo, sono in grado di gestire sequenze di dati di lunghezza variabile, grazie all'utilizzo del feedback. In secondo luogo, sono in grado di gestire sequenze di dati con dipendenze temporali complesse, grazie alla memoria interna del livello ricorrente. Infine, sono in grado di gestire anche sequenze di dati con rumore o distorsioni, grazie alla capacità di generalizzazione della rete.

In sintesi, le RNN sono un tipo di rete neurale artificiale che si basa sull'utilizzo del feedback per elaborare sequenze di dati. Grazie alla loro capacità di gestire sequenze di dati di lunghezza variabile e con dipendenze temporali complesse, le RNN sono utilizzate in diversi campi per risolvere problemi di previsione, di classificazione e di intelligenza artificiale.

3.3.3 RETI NEURALI DENSAMENTE CONNESSE

Le reti neurali densamente connesse sono costituite da una serie di unità di elaborazione chiamate neuroni artificiali o perceptron. I neuroni perceptron sono organizzati in strati, con ogni strato che comunica con il successivo tramite

3.3. APPROCCI BASATI SULL'APPRENDIMENTO AUTOMATICO NEL RICONOSCIMENTO DI GESTURE E AZIONI UMANE

connessioni pesate. La disposizione delle connessioni tra i neuroni è tale che ogni neurone in un determinato strato è connesso a tutti i neuroni dello strato successivo. Ogni neurone in una rete neurale densamente connessa accetta input ponderati dai neuroni nel livello precedente, applica una funzione di attivazione non lineare e produce un output che viene passato ai neuroni nel livello successivo. Questo processo di propagazione dell'input attraverso la rete viene chiamato inoltro (forward pass).

Le connessioni tra i neuroni hanno dei pesi associati, che rappresentano l'importanza relativa di ciascuna connessione nell'influenzare l'output finale della rete. Durante l'addestramento della rete, questi pesi vengono ottimizzati per ridurre l'errore tra l'output previsto della rete e l'output desiderato. Questo processo di ottimizzazione viene spesso realizzato mediante l'algoritmo di retropropagazione dell'errore (backpropagation), che calcola l'errore nella rete e aggiorna i pesi delle connessioni in base a tale errore. Una rete neurale densamente connessa, come mostrato in figura 3.4, è composta da tre tipi di strati: lo strato di input, uno o più strati nascosti e lo strato di output.

Lo strato di input, Input Layer in figura 3.4, riceve i dati in ingresso alla rete e rappresenta le variabili indipendenti del problema. Il numero di neuroni in questo strato dipende dal numero di caratteristiche o attributi che si desidera utilizzare come input per la rete, nella rete descritta dalla figura 3.4 gli attributi in input saranno cinque, come i neuroni del primo livello.

Gli strati nascosti, Hidden Layer in figura 3.4, rappresentano il cuore della rete neurale densamente connessa. Ogni strato nascosto è composto da un insieme di neuroni artificiali che elaborano l'input dai neuroni dello strato precedente. Questi strati lavorano insieme per estrarre caratteristiche rilevanti dai dati di input e apprendere rappresentazioni interne complesse.

Lo strato di output, Output Layer in figura 3.4, produce l'output finale della rete. Il numero di neuroni in questo strato dipende dal tipo di problema che si sta affrontando. Ad esempio, per un problema di classificazione binaria, potrebbe esserci un singolo neurone nell'output che rappresenta la probabilità di appartenenza a una delle due classi. Mentre per un problema di classificazione multi-classe, il numero di neuroni nell'output corrisponde al numero di classi, nella rete descritta dalla figura 3.4 i possibili output saranno quindi quattro, come i neuroni dell'ultimo livello.

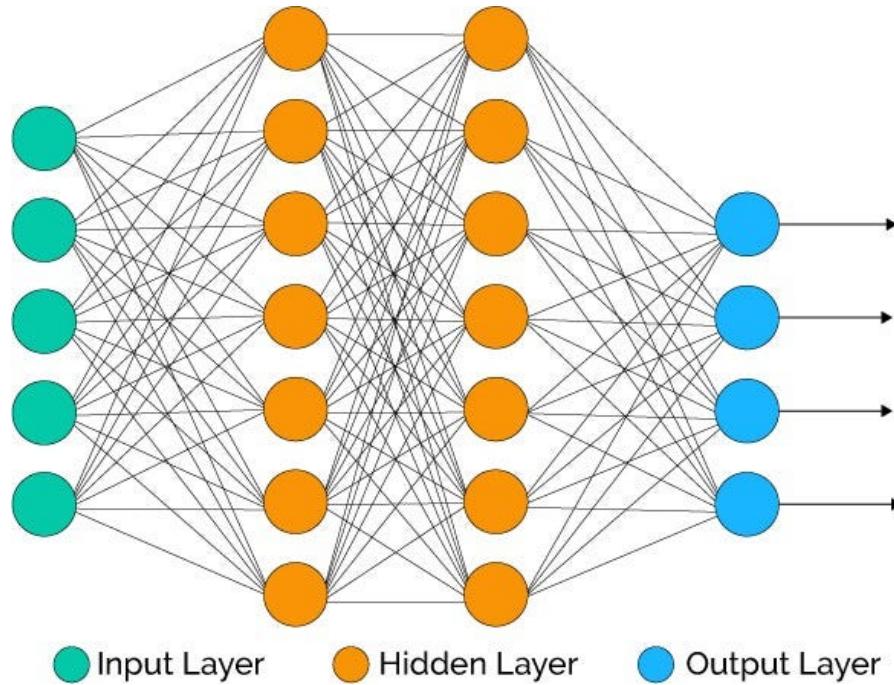


Figura 3.4: Architettura di una Rete Densamente Connessa.

Le reti neurali densamente connesse hanno dimostrato di essere molto versatili e sono state utilizzate in una vasta gamma di applicazioni. Alcuni esempi di campi di utilizzo sono il riconoscimento di immagini, il riconoscimento vocale e la traduzione automatica di testi.

4

Setup sperimentale ed acquisizione del dataset

In questo capitolo viene presentata l'attività sperimentale svolta presso l'IAS-Lab del Dipartimento di Ingegneria dell'Informazione dell'Università di Padova, che ha come obiettivo lo sviluppo di un sistema di riconoscimento di azioni basato su dati acquisiti con sensori inerziali. Viene quindi descritta la tecnologia utilizzata nell'esperienza e il set-up impiegato per la raccolta dei dati necessari alla creazione del dataset.

4.1 TECNOLOGIA E SET-UP IMPIEGATI PER LA CREAZIONE DEL DATASET

Per la raccolta dei dati necessari alla creazione del dataset, sono stati utilizzati tre MTw Awinda Development kit¹.

Ciascuno dei quali dotato di:

- Sei sensori inerziali Xsens MTw (Miniature Technology wireless) Awinda
- Awinda station
- Chiavetta USB Awinda
- Set di cinturini in velcro per tutto il corpo
- Software XSense MVN

¹<https://www.movella.com/products/wearables/xsens-mtw-awinda>

4.1. TECNOLOGIA E SET-UP IMPIEGATI PER LA CREAZIONE DEL DATASET

Il kit di sviluppo Mtw Awinda, figura 4.1, contiene l'attrezzatura necessaria per l'acquisizione dei dati..

Gli Xsens MTw Awinda, figura 4.2, sono degli strumenti di misura inerziale di dimensioni ridotte ma estremamente efficienti, composti da accelerometri lineari 3D, giroscopi 3D, magnetometri 3D ed un barometro [22].

Il design dei sensori MTw è stato ideato per agevolarne l'utilizzo, grazie alla presenza di un pratico patch in velcro sul retro della custodia, che consente di fissare facilmente il sensore ai cinturini del corpo.

La stazione Awinda, figura 4.3, rappresenta un importante dispositivo di controllo e monitoraggio della ricezione dei dati trasmessi dai dispositivi MTw ed è in grado di ricaricare fino a 6 MTw contemporaneamente.

La stazione Awinda è progettata per ricevere dati wireless da un massimo di 32 MTw, garantendo una gestione efficiente della ricezione e dell'elaborazione dei dati ricevuti.

La chiavetta USB Awinda, figura 4.4, offre le stesse funzionalità della stazione Awinda, permette infatti la ricezione dei dati wireless sincronizzati da tutti gli MTw connessi, inoltre è necessaria per utilizzare il software XSens MVN che elabora i dati raccolti dai sensori. Anche la chiavetta USB è in grado di ricevere dati da un massimo di 32 MTw.

Il kit di sviluppo Awinda è fornito di un set di cinturini, figura 4.5, ed una maglietta dotati di strap e parti in velcro per fermare sul corpo i sensori. Lo strato superiore di ciascun cinturino è infatti realizzato in velcro in modo tale che i sensori possano aderire e rimanere fermi durante l'acquisizione.

Il software XSens MVN, figura 4.6, è composto dal Motion Tracker Manager e da un kit di sviluppo software. Il Motion Tracker Manager serve per visualizzare e registrare i dati.



Figura 4.1: Mtw Awinda Dev. kit.



Figura 4.2: Xsens MTw Awinda.



Figura 4.3: Awinda Station.



Figura 4.4: Chiavetta USB Awinda.



Figura 4.5: Set cinturini in velcro.

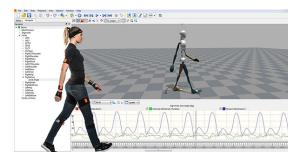


Figura 4.6: Software XSens MVN.

La tecnologia Xsens rappresenta una soluzione avanzata per il monitoraggio dei dati di movimento del corpo. Essa consiste in una tuta e dei cinturini per fissare i sensori inerziali, che vengono posizionati su varie parti del corpo per raccogliere dati di accelerazione, velocità angolare e orientamento angolare in tre dimensioni. Grazie alla tecnologia wireless Awinda, i dati raccolti dai sensori possono essere trasmessi in tempo reale alla stazione e analizzati mediante tecniche di filtraggio e segmentazione.

La procedura di acquisizione dati prevede che l'utente esegua le azioni pre-definite per raccogliere dati rappresentativi dei movimenti compiuti dal corpo. Il processore incorporato nei sensori gestisce il campionamento, il buffering, la calibrazione e l'integrazione dei dati inerziali, assicurando un alto livello di precisione e affidabilità.

Una delle caratteristiche uniche del MTw è il protocollo radio brevettato Awinda, che garantisce la sincronizzazione temporale di fino a 20 MTw nella rete wireless con una precisione di 10 microsecondi. Tale protocollo assicura l'accuratezza del tracciamento del movimento 3D anche se i dati vengono temporaneamente persi durante la trasmissione radio, mantenendo un uso molto efficiente della limitata larghezza di banda disponibile.

Grazie alla natura completamente wireless dei sensori MTw questa tecnologia trova applicazione in diverse aree, migliorando la velocità di applicazione

4.2. DATASET

dei sistemi di tracciamento del movimento sui soggetti di test o sui pazienti, senza la necessità di ambienti simulati, questo anche grazie al fatto che la tecnologia dei sensori inerziali non è influenzata dalle condizioni di illuminazione e da altri fattori climatici.

4.2 DATASET

Al fine di sviluppare e validare una rete neurale nel riconoscimento di azioni, è stato creato un dataset contenente otto diverse azioni, alcune delle quali molto simili tra loro e altre molto diverse. La selezione di tali azioni è stata fatta con l'obiettivo di rendere l'analisi dei risultati ottenuti dal riconoscimento più interessante, valutando l'accuratezza massima ottenibile nel riconoscimento delle azioni più simili confrontata all'accuratezza ottenuta nel riconoscimento di quelle meno simili tra loro.

Ad esempio, si è scelto di includere nel dataset le due azioni "Camminare avanti" (Azione 2) e "Camminare all'indietro" (Azione 6) che rappresentano un notevole livello di somiglianza. Queste due azioni hanno una natura simile per quanto riguarda il movimento del corpo ma differiscono nella direzione del movimento rispetto all'orientamento del corpo.

Al contrario, due azioni scelte che rappresentano un minor grado di somiglianza sono "Raccogliere una bottiglia" (Azione 4) e "Simulare nuotata a stile" (Azione 5). Queste azioni consistono in movimenti del corpo completamente diversi, la prima azione infatti coinvolge principalmente gli arti superiori e non presenta uno spostamento del soggetto, al contrario, la seconda azione prevede una camminata iniziale verso la bottiglia, quindi uno spostamento del soggetto dal punto di partenza e il movimento degli arti inferiori per la flessione al fine di raccogliere la bottiglia.

Infine, un'altra coppia di azioni ben semanticamente definite, seppur molto simili nei movimenti e parti del corpo coinvolte nell'azione, che rappresenta quindi un caso di studio particolarmente interessante, è costituita da "Eseguire degli squat" (Azione 3) e "Saltare" (Azione 8). Entrambe le azioni prevedono un movimento verticale senza spostarsi lateralmente, ma si differenziano nel modo in cui viene effettuato il movimento. Durante l'esecuzione degli squat, si esegue una flessione delle ginocchia e delle anche operando uno spostamento

verso il basso per poi ritornare in posizione eretta, il salto coinvolge invece un movimento verticale che spinge il corpo verso l'alto.

Il dataset che si è scelto di realizzare è composto dalle seguenti otto azioni: 1) posizione ferma; 2) camminare avanti; 3) eseguire degli squat; 4) raccogliere una bottiglia; 5) simulare nuotata a stile; 6) camminare all'indietro; 7) eseguire dei piegamenti; 8) saltare.



Figura 4.7: Azioni del dataset raccolto presso lo IAS-Lab.

Le registrazioni delle azioni necessarie alla formazione del dataset si sono basate su tre soggetti di differente altezza, in modo tale da rendere il dataset il più esaustivo possibile.

Per ogni soggetto sono state registrate cinque ripetizioni da cinque secondi ciascuna, per ognuna delle otto azioni.

Dunque la dimensione del dataset, essendo composto da cinque ripetizioni per otto azioni di tre soggetti, è di $5 \times 8 \times 3 = 120$ acquisizioni.

4.3. ACQUISIZIONE DATI



Figura 4.8: Soggetti partecipanti alla creazione del dataset.

4.3 Acquisizione dati

Per iniziare le acquisizioni, è stato necessario preparare la postazione di raccolta dati, collegando la stazione di ricezione Awinda e la chiavetta USB Awinda al PC utilizzato per la raccolta dei dati. Inoltre, è stato fondamentale fornire al software XSens MVN alcune informazioni sul soggetto che indossava la tuta con i sensori posizionati, informazioni quali l'altezza del soggetto e la lunghezza del suo piede, in modo tale da ottenere una ricostruzione del corpo il più accurata e precisa possibile.

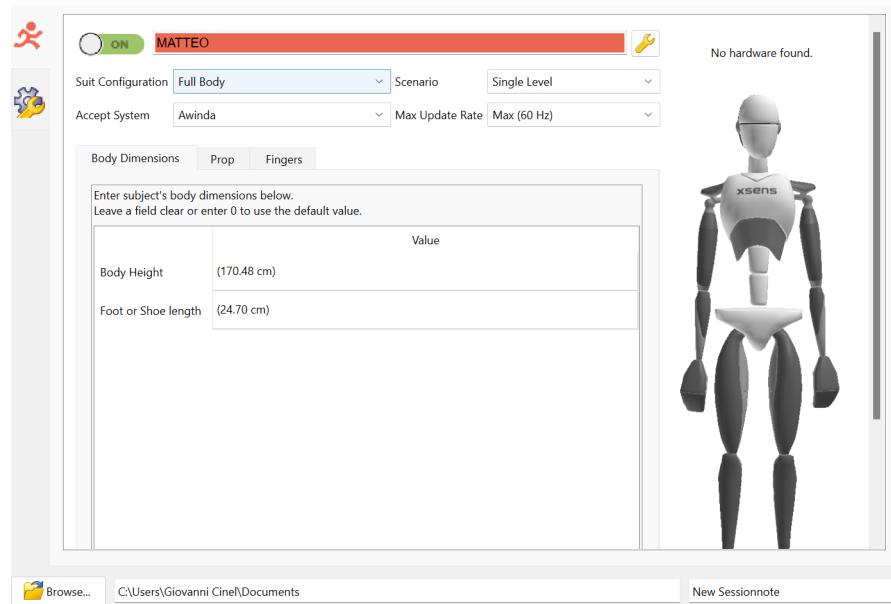


Figura 4.9: Configurazione per le acquisizioni del terzo soggetto.

Per le acquisizioni sono stati utilizzati 17 sensori, disposti nel seguente modo:

- | | | |
|--------------------|---------------------|---------------------|
| 1. Testa | 7. Braccio sinistro | 13. Coscia sinistra |
| 2. Spalla destra | 8. Polso destro | 14. Gamba destra |
| 3. Spalla sinistra | 9. Polso sinistro | 15. Gamba sinistra |
| 4. Bacino | 10. Mano destra | 16. Piede destro |
| 5. Sterno | 11. Mano sinistra | 17. Piede sinistro |
| 6. Braccio destro | 12. Coscia destra | |

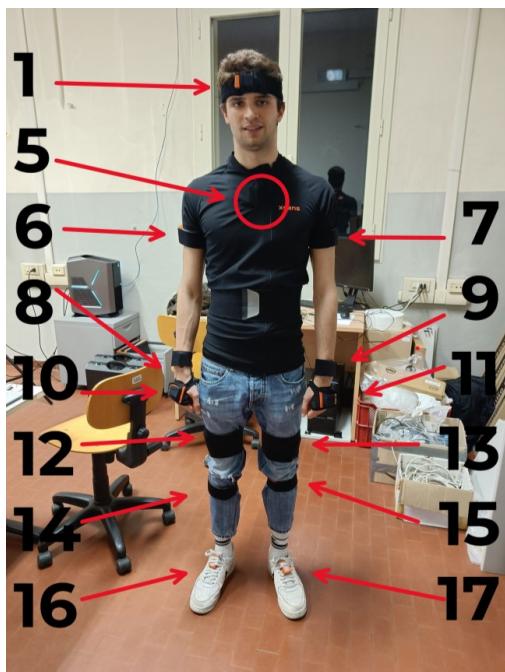


Figura 4.10: Sensori nella regione anteriore.



Figura 4.11: Sensori nella regione posteriore.

Una volta completata la configurazione del software, è stato possibile procedere con la calibrazione dei sensori.

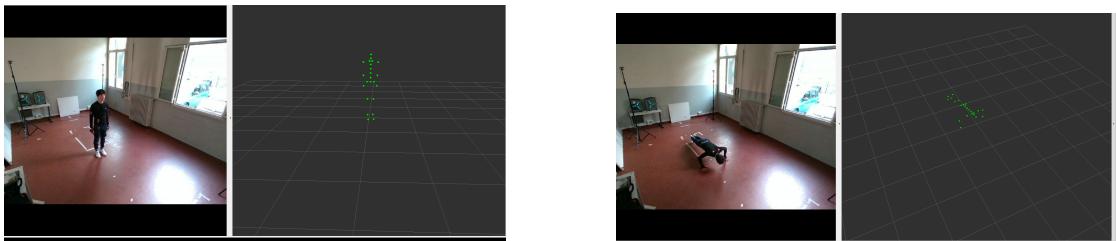
La calibrazione dei sensori è un processo importante per garantire la precisione e l'affidabilità delle misurazioni effettuate da questi dispositivi. Per iniziare, è necessario posizionare i sensori sul corpo del soggetto, utilizzando le cinghie in dotazione per assicurarsi che siano ben fissati e che non si spostino durante il movimento. Una volta che i sensori sono posizionati correttamente è necessario eseguire il software fornito da Xsens per avviare la calibrazione, seguendo le indicazioni vocali, riguardanti la posizione da assumere e i movimenti da svolgere,

4.3. ACQUISIZIONE DATI

forniti dal software. In questo modo il programma può operare una ricostruzione del soggetto basata su un modello biomeccanico del corpo umano stimando la distanza dei sensori e sfruttando i dati riguardanti l'altezza e la lunghezza del piede del soggetto precedentemente forniti al software. Il risultato finale è un sistema di sensori altamente preciso e affidabile, che offre dati di movimento accurati e una rappresentazione precisa del corpo umano del soggetto, espressa come uno scheletro di 23 giunti e 22 link. È necessario rieseguire la calibrazione in modo periodico o in caso di dissestamento della posizione dei sensori, per mantenere alta la precisione durante la raccolta dati.

Completata la calibrazione è stato possibile iniziare le acquisizioni.

Il software XSens si occupa di ricostruire lo scheletro del soggetto calcolando il movimento dei giunti del corpo in base alle misure dei sensori ed utilizzando un modello cinematico del corpo umano, lo scheletro non viene ricostruito mostrando la posizione nel tempo dei 17 sensori, ma mostrando la posizione e i movimenti di 23 giunti del corpo, calcolati grazie ai dati trasmessi dai sensori.



(a) Posizione dei giunti nell'azione 1

(b) Posizione dei giunti nell'azione 7

Figura 4.12: Posizione del soggetto descritta dai giunti.

I 23 giunti con i quali viene rappresentato il soggetto sono i seguenti:

- | | | |
|------------------|--------------------------|------------------------------|
| 1. Bacino | 9. Braccio destro | 17. Gamba destra |
| 2. Vertebra L5 | 10. Avambraccio destro | 18. Caviglia destra |
| 3. Vertebra L3 | 11. Mano destra | 19. Punta del piede destro |
| 4. Vertebra T12 | 12. Spalla sinistra | 20. Coscia sinistra |
| 5. Sterno | 13. Braccio sinistro | 21. Gamba sinistra |
| 6. Collo | 14. Avambraccio sinistro | 22. Caviglia sinistra |
| 7. Testa | 15. Mano sinistra | 23. Punta del piede sinistro |
| 8. Spalla destra | 16. Coscia destra | |

CAPITOLO 4. SETUP SPERIMENTALE ED ACQUISIZIONE DEL DATASET

Durante il processo di acquisizione dati è stato eseguito un nodo ROS²(Robot Operating System) che ha registrato, ad una frequenza di 60Hz, la posizione e l'orientamento di ciascun giunto del sistema.

Tali informazioni sono state salvate in file rosbag, avente estensione .bag, e archiviate all'interno di apposite strutture dati.

²<https://www.ros.org/>

5

Sviluppo della rete neurale

In questo capitolo vengono illustrate le fasi di elaborazione dei dati e di allenamento delle reti neurali per la classificazione delle azioni. Infine, verranno presentati i risultati dell'apprendimento delle reti neurali, e la conseguente affidabilità del modello, ottenuti con il dataset sviluppato.

Il tool di machine learning utilizzato per allenare un algoritmo al riconoscimento delle azioni scelte con i dati ottenuti attraverso le acquisizioni è Edge Impulse¹. Il tool di machine learning scelto, Edge Impulse, richiede i dati in formato diverso dalle bag di ROS, formato nel quale sono stati salvati durante le acquisizioni, per poter quindi procedere all'utilizzo dei dati raccolti per allenare l'algoritmo di riconoscimento di azioni, è stato necessario operare una conversione dei dati raccolti dai sensori e salvati in file rosbag in file di formato adatto ad essere elaborati da Edge Impulse, ovvero in formato Comma Separated Values, avente estensione .csv.

5.1 CONVERSIONE DEI DATI

Al fine di operare la conversione dei dati raccolti in un formato utile per l'elaborazione è stato sviluppato il codice riportato in Appendice A.

L'implementazione del codice descritto richiede che l'utente fornisca in input il percorso della cartella contenente i file con estensione .bag da cui si desidera estrarre i dati da analizzare. Il programma legge i file presenti nella cartella, e per ciascuno di essi produce come output un file in formato Comma Separated

¹<https://www.edgeimpulse.com/>

5.2. ALLENAMENTO DELLA RETE

Values contenente per ogni timestamp i dati relativi alla posizione e all’orientamento dei ventitre giunti del sistema presenti nel topic SkeletonQuaternion dei messaggi ROS salvati nei file .bag. Il formato dei file .csv creati include una riga di intestazione contenente i nomi dei campi, tra cui il timestamp e le posizioni e gli orientamenti dei giunti. Per ogni timestamp vengono salvati i dati in una riga del file, separando i valori di posizione e orientamento di ogni giunto con una virgola.

5.2 ALLENAMENTO DELLA RETE

I dati sono stati caricati su Edge Impulse in file di formato CSV prodotti dal codice riportato in Appendice A.

Per lo sviluppo dell’algoritmo di riconoscimento è stato diviso il dataset in due parti: un training set, costituito dal 80% dei dati, e un test set, costituito dal restante 20%. I dati sono stati ripartiti nel seguente modo, per ogni azione quattro delle cinque acquisizioni di ciascun soggetto sono andate a comporre il training set, al test set invece è stata assegnata l’acquisizione restante per azione di ciascun soggetto. Il training set è stato utilizzato per allenare l’algoritmo di riconoscimento, cioè per far sì che fosse in grado di riconoscere le azioni del dataset e di formulare previsioni su nuovi dati. Il test set, invece, è stato utilizzato per verificare l’accuratezza delle previsioni effettuate dal classificatore una volta ottenuto il modello. In questo modo è stato possibile misurare le prestazioni dell’algoritmo su dati che non erano stati utilizzati per l’addestramento, ottenendo quindi una valutazione indipendente della sua capacità di generalizzazione.

Preparati i dati per lo sviluppo del modello è stato necessario scegliere il metodo di apprendimento.

Si è impostato il tool di machine learning per utilizzare la classificazione come modalità di apprendimento. Questo tipo di apprendimento automatico si basa sull’esistenza di un insieme di etichette di output note, e ha come obiettivo l’allenamento di un algoritmo in modo tale che sia capace di predire le etichette di output in relazione ai dati di input. Il processo di allenamento prevede l’utilizzo di un training set costituito da coppie ordinate di elementi di input e le relative etichette di output. Tale processo è finalizzato a minimizzare l’errore di predizione tra le etichette di output calcolate e quelle effettive.

Nel contesto di Edge Impulse, viene sviluppata una rete neurale densamente connessa per la classificazione dei dati. La struttura di base della rete consiste in uno o più strati di neuroni, chiamati layer, che sono connessi densamente con gli strati adiacenti. Ogni neurone in uno strato è collegato a tutti i neuroni dello strato successivo, inclusi quelli dello strato di output. Durante l'addestramento, la rete regola i pesi delle connessioni tra i neuroni per migliorare la precisione del classificatore.

Negli esperimenti svolti (sezione 5.2.1), si sono allenate varie configurazioni di rete neurale per trovare in modo empirico la miglior configurazione di parametri per il problema in esame. Sono stati trovati quindi i valori del tasso di apprendimento, del numero di cicli di allenamento e della dimensione del set di validazione necessari per ottenere un modello con l'accuracy massimizzata.

Il tasso di apprendimento (learning rate) è un parametro che viene utilizzato in algoritmi di apprendimento automatico per regolare la velocità con cui il modello impara dai dati durante la fase di addestramento. In pratica, il learning rate indica di quanto modificare i pesi delle connessioni tra i neuroni durante l'aggiornamento dei parametri del modello. Un learning rate troppo basso rallenta l'apprendimento e può portare a una convergenza lenta o addirittura a una situazione di stallo, mentre un learning rate troppo alto può causare oscillazioni o una convergenza troppo rapida a soluzioni subottimali.

La scelta corretta del learning rate è quindi importante per ottenere un modello preciso e ben addestrato.

Il numero di cicli di allenamento (o epoch) indica il numero di volte che l'algoritmo viene eseguito sull'intero set di dati durante la fase di allenamento. In pratica, ogni epoca consiste in una passata di tutti i dati attraverso il modello, durante la quale vengono aggiornati i pesi delle connessioni tra i neuroni sulla base della differenza tra la risposta prevista dal modello e la risposta corretta.

In generale, più epoch vengono eseguite, più il modello ha la possibilità di imparare dai dati e migliorare la sua capacità predittiva. Tuttavia, questo può comportare un rischio di overfitting, ovvero una situazione in cui il modello adatta troppo la sua rappresentazione ai dati di addestramento, perdendo la capacità di generalizzare su nuovi dati.

La scelta del numero di epoch dipende quindi dalle caratteristiche specifiche del problema e del dataset di addestramento, e può richiedere una certa sperimentazione e ottimizzazione.

La dimensione del set di validazione indica la percentuale di dati apparte-

5.2. ALLENAMENTO DELLA RETE

nenti al training set utilizzati per valutare le prestazioni del modello addestrato sui restanti dati del set di allenamento. Il set di validazione è quindi una parte del set di allenamento, i cui dati non vengono utilizzati per addestrare il modello, ma solo per valutarne l'accuratezza durante l'allenamento e individuare eventuali problemi di overfitting. Il set di validazione viene quindi utilizzato per verificare e valutare i progressi, nella classificazione delle azioni operata dal modello, durante la fase di training. Anche la scelta del set di validazione è importante, poiché deve rappresentare in modo equilibrato il problema e le sue possibili variazioni, Edge Impulse non permette di scegliere quali dati inserire nel set di validazione, si limita a fornire la possibilità di indicare la percentuale dei dati del training set da destinare al validation set.

I valori che ottimizzano la precisione e l'accuratezza del modello sono risultati essere i seguenti:

- Tasso di apprendimento : 0.0005
- Numero di cicli di allenamento : 100
- Dimensioni set di validazione : 20

L'allenamento della rete neurale descritta dai parametri della configurazione 29, configurazione descritta nella tabella 5.1, ha prodotto un modello con i seguenti risultati in fase di validazione:



Confusion matrix (validation set)

	1	2	3	4	5	6	7	8
1	100%	0%	0%	0%	0%	0%	0%	0%
2	0%	75%	0%	25%	0%	0%	0%	0%
3	0%	0%	100%	0%	0%	0%	0%	0%
4	7.1%	7.1%	7.1%	64.3%	0%	0%	0%	14.3%
5	0%	0%	0%	0%	100%	0%	0%	0%
6	0%	0%	0%	0%	0%	100%	0%	0%
7	0%	0%	0%	0%	0%	0%	100%	0%
8	0%	0%	0%	0%	0%	0%	0%	100%
F1 SCC	0.94	0.82	0.93	0.69	1.00	1.00	1.00	0.93

Figura 5.1: Performance ottenute con il set di validazione.

Dalla figura 5.1, che illustra i risultati ottenuti in validazione, è possibile osservare la percentuale di acquisizioni del validation set classificate in modo corretto per ciascuna delle otto azioni del dataset, e più in specifico, per ogni azione in input la percentuale di acquisizioni che il modello ha classificato come ciascuna delle possibili azioni in output. Le righe rappresentano la label corretta dell'acquisizione da classificare, le colonne descrivono le possibili risposte. Ad esempio il valore 25% presente nella seconda riga e quarta colonna, ci dice che delle acquisizioni del validation set che contengono l'azione 2 (Camminare avanti) l'algoritmo di riconoscimento ne ha classificate il 25% come azione 4 (Raccogliere una bottiglia). È possibile osservare che l'algoritmo non ha riconosciuto perfettamente solo le azioni 2 e 4. In particolare le acquisizioni contenenti l'azione 2 sono state erroneamente classificate come azioni 4 il 25% delle volte, e le acquisizioni contenenti l'azione 4 sono state classificate in modo errato come

5.2. ALLENAMENTO DELLA RETE

azione 1 (Posizione ferma) nel 7.1% dei casi, come azione 2 nel 7.1% dei casi, come azione 3 (Eseguire degli squat) nel 7.1% dei casi e come azione 8 (Saltare) nel 14.3% dei casi, ciò ha comportato una corretta classificazione dell'azione 4 solo nel 64.3% dei casi. La difficoltà nella corretta classificazione delle acquisizioni contenenti l'azione 4 è dovuta alla somiglianza che presenta lo svolgimento dell'azione 4 con le altre azioni con le quali viene confusa. Infatti l'esecuzione dell'azione 4 prevede una iniziale camminata verso la bottiglia, che comporta un possibile motivo di errata classificazione nell'azione 2, per poi proseguire con una flessione verso il basso, movimento che ricorda uno squat e possibile causa di errata classificazione nell'azione 3. L'azione si conclude raccogliendo la bottiglia e ritornando in posizione eretta, possibile fonte di errata classificazione nell'azione 8, visto lo spostamento del corpo del soggetto verso l'alto. Complessivamente i risultati sono comunque molto soddisfacenti, presentano infatti un'accuracy del 89.5%.

Data explorer (full training set) ?

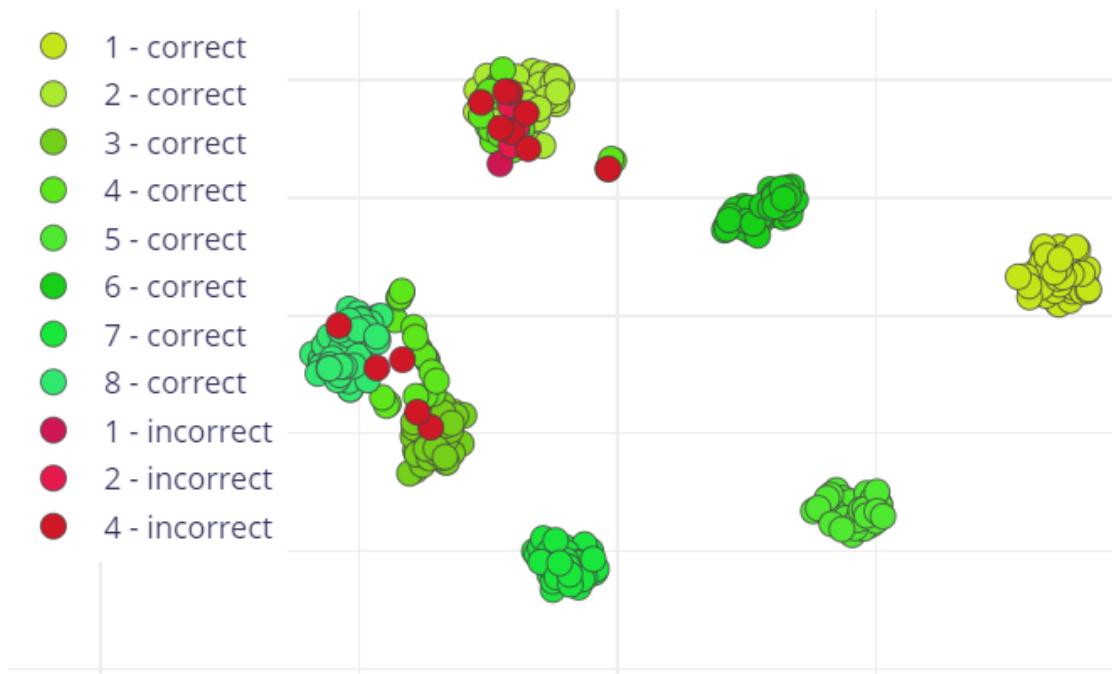


Figura 5.2: Elementi del training set classificati dalla rete neurale.

Con il modello di cui sono stati presentati i risultati in validazione, in figura 5.1, sono stati classificati tutti gli elementi appartenenti al training set, non solo

il 20% del validation set, ottenendo i risultati mostrati in figura 5.2. In verde è possibile osservare gli elementi del training set classificati correttamente mentre in rosso quelli classificati in modo errato dal modello.

5.2.1 RISULTATI OTTENUTI DALLE RETI NEURALI ALLENATE

Vengono riportati nella tabella 5.1 i risultati ottenuti nei vari tentativi che hanno portato a trovare la rete neurale migliore per il riconoscimento delle azioni del dataset del caso di studio presentato. Ogni riga rappresenta una configurazione di architettura delle reti, dove in colonna sono riportati: il tasso di apprendimento (T.A.), il numero di cicli di allenamento (C.A.), il numero di livelli della rete neurale (n° lvl), il numero di neuroni del primo livello (lvl 1), il numero di neuroni del secondo livello (lvl 2), il numero di neuroni del terzo livello (lvl 3), l'accuracy nel validation set (Validazione) e l'accuracy nel test set (Test).

Nelle reti neurali configurate con un numero di strati pari a due, si osserva la presenza di una variabile simbolica, rappresentata dalla lettera "x", nel contesto relativo alla quantità di neuroni assegnati al terzo livello. Tale simbolo evoca l'omissione del parametro indicato, sottolineando la sua non utilizzazione nell'architettura della rete.

5.2. ALLENAMENTO DELLA RETE

	T.A.	C.A	n° lvl	lvl 1	lvl 2	lvl 3	Validazione	Test
Configurazione 1	0.0001	500	2	20	10	x	89.5%	81.05%
Configurazione 2	0.0001	500	2	30	20	x	86.8%	85.26%
Configurazione 3	0.0001	500	2	30	10	x	89.5%	86.32%
Configurazione 4	0.0001	500	2	50	30	x	82.9%	82.11%
Configurazione 5	0.0001	500	3	50	30	20	89.5%	84.21%
Configurazione 6	0.0001	500	3	40	20	10	89.5%	87.37%
Configurazione 7	0.0001	500	3	40	30	10	90.8%	86.32%
Configurazione 8	0.0001	500	3	40	30	20	86.8%	85.26%
Configurazione 9	0.0001	500	3	100	90	70	89.5%	85.26%
Configurazione 10	0.0001	500	2	100	90	x	86.8%	85.26%
Configurazione 11	0.0002	250	2	20	10	x	88.2%	84.21%
Configurazione 12	0.0002	250	2	30	20	x	88.2%	91.56%
Configurazione 13	0.0002	250	2	30	10	x	90.8%	85.26%
Configurazione 14	0.0002	250	2	50	30	x	85.5%	87.37%
Configurazione 15	0.0002	250	3	50	30	20	85.5%	83.16%
Configurazione 16	0.0002	250	3	40	20	10	89.5%	81.05%
Configurazione 17	0.0002	250	3	40	30	10	88.2%	87.37%
Configurazione 18	0.0002	250	3	40	30	20	86.8%	84.21%
Configurazione 19	0.0002	250	2	100	90	80	89.5%	87.37%
Configurazione 20	0.0002	250	2	100	90	x	86.8%	90.53%
Configurazione 21	0.0005	100	2	20	10	x	84.2%	66.32%
Configurazione 22	0.0005	100	2	30	20	x	86.8%	82.11%
Configurazione 23	0.0005	100	2	30	10	x	84.2%	78.95%
Configurazione 24	0.0005	100	2	50	30	x	86.8%	87.37%
Configurazione 25	0.0005	100	3	50	30	20	86.8%	84.21%
Configurazione 26	0.0005	100	3	40	20	10	92.1%	83.16%
Configurazione 27	0.0005	100	3	40	30	10	86.8%	80.00%
Configurazione 28	0.0005	100	3	40	30	20	88.2%	87.37%
Configurazione 29	0.0005	100	3	100	90	80	89.5%	91.58%
Configurazione 30	0.0005	100	2	100	90	x	89.5%	90.53%

Tabella 5.1: Risultati dell’allenamento delle reti neurali.

5.3 RISULTATI OTTENUTI

Valutando i risultati conseguiti nella fase di sperimentazione si è scelto il modello ottenuto allenando la rete neurale con i parametri descritti dalla configurazione 29, presente nella tabella 5.1, modello che ha ottenuto i risultati migliori di accuratezza nella classificazione degli elementi del test set, classificazione che ha prodotto i risultati riportati in figura 5.3.

In conclusione, l'algoritmo di riconoscimento sviluppato ha dimostrato di avere un'ottima capacità di previsione, raggiungendo un'accuracy del 91.58% sul test set utilizzato per la valutazione delle prestazioni, dimostrando quindi una buona capacità di generalizzazione nel riconoscere il set di azioni scelto su nuovi dati (ovvero, non usati per l'allenamento).

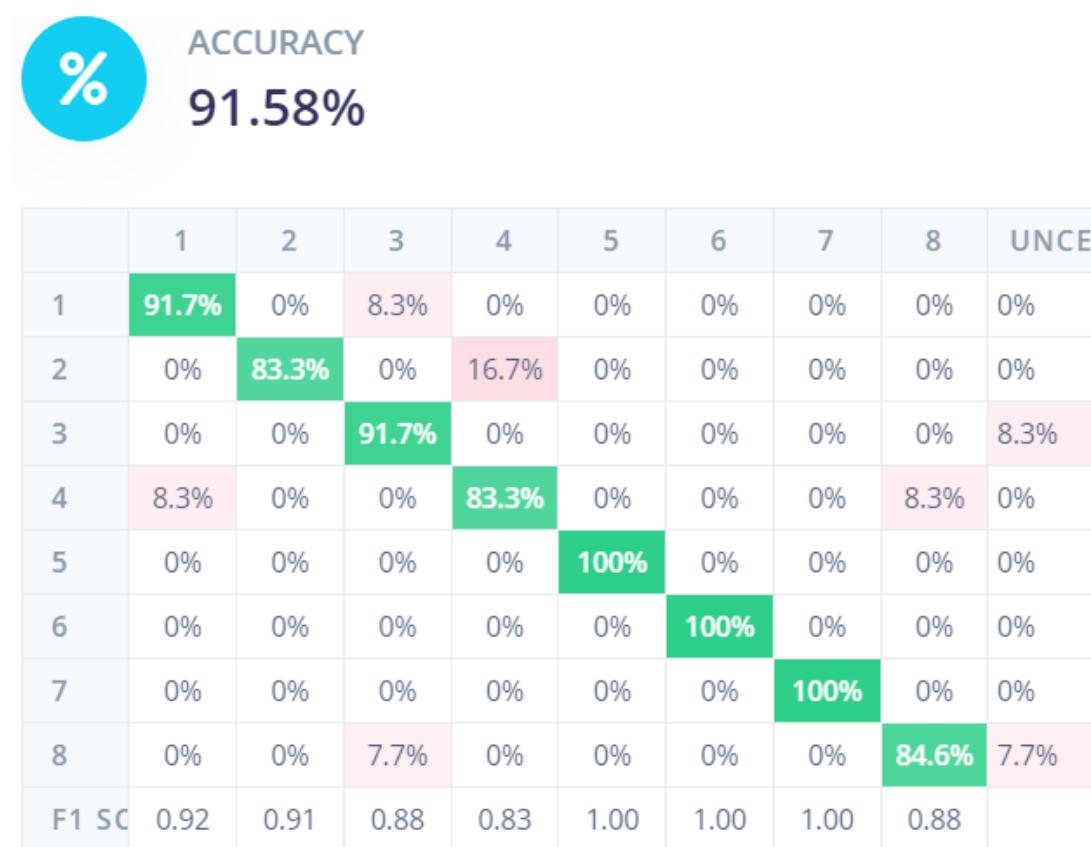


Figura 5.3: Risultati ottenuti con il test set.

Dalla figura 5.3, che illustra i risultati ottenuti, è possibile osservare la probabilità di classificare in modo corretto ciascuna delle otto azioni del dataset, e più

5.3. RISULTATI OTTENUTI

in specifico, per ogni azione in input la probabilità che venga classificata come ciascuna delle possibili azioni in output.

Le righe rappresentano la label corretta dell'azione da classificare, le colonne descrivono le possibili risposte con la probabilità indicata.

Ad esempio il valore 8.3% presente nella quarta riga, prima colonna ci dice che dall'analisi di un'acquisizione che contiene l'azione 4 (Raccogliere una bottiglia) la probabilità che l'algoritmo di riconoscimento la classifichi come azione 1 (Posizione ferma) è del 8.3%.

Viene riportata per ogni azione anche la probabilità con la quale l'algoritmo sarà incerto nella risposta (nona colonna a partire da sinistra).

È possibile osservare che l'algoritmo trova le maggiori difficoltà nel riconoscere le azioni 2 e 4, in particolare nel 16.7% dei casi classifica erroneamente l'azione 2 come azione 4, mentre la classificazione dell'azione 4 presenta l'8.3% di probabilità di essere classificata erroneamente come azione 1 e sempre l'8.3% di probabilità di essere classificata erroneamente come azione 8.

La difficoltà nel classificare correttamente le due azioni considerate, quindi l'azione 2 e l'azione 4, è dovuta alla forte somiglianza ad altre azioni. Nell'eseguire l'azione 4, ovvero raccogliere una bottiglia, il soggetto cammina verso la bottiglia per poi raccoglierla, la prima parte dell'azione è molto simile all'azione 2, ovvero la camminata avanti. Nella parte finale invece il soggetto si piega per raccogliere da terra la bottiglia, compiendo un movimento simile allo squat, ovvero l'azione 8.

Questa somiglianza risulta essere un limite per l'algoritmo di classificazione, che non permette il corretto etichettamento, nel 100% dei casi, di queste azioni.

6

Conclusione e sfide aperte nel campo del riconoscimento di gesture e human action

6.1 CONCLUSIONE

In questa tesi si è affrontato il tema del riconoscimento di azioni. L'esperienza presentata si è visto essere composta da diverse fasi, dalla costruzione di un dataset di otto azioni, all'implementazione di un algoritmo di riconoscimento di queste.

La creazione del dataset ha richiesto un'attenzione particolare nella scelta delle azioni da includere, in modo tale da poter analizzare e confrontare i risultati ottenuti con azioni a diversi livelli di somiglianza. Sono stati inoltre introdotti degli accorgimenti, come l'utilizzo di diversi soggetti nelle acquisizioni, per rendere il dataset il più variegato possibile.

Sulla base dei risultati ottenuti, è possibile affermare che l'algoritmo ha dimostrato un'ottima capacità di riconoscimento delle azioni più distinte, come atteso, mentre ha mostrato una leggera limitazione nel riconoscere le azioni più simili tra loro. Tuttavia, nonostante questa limitazione, l'algoritmo ha comunque fornito risultati globalmente superiori alle aspettative e ha raggiunto un alto livello di affidabilità nel riconoscimento delle azioni selezionate.

L'accuracy del modello di riconoscimento ottenuta (91.58%) è ottima e garantisce un riconoscimento affidabile delle azioni, tuttavia, esistono ancora sfide e possibili sviluppi futuri nella ricerca del riconoscimento di azioni, in particolare

6.2. SFIDE APERTE E PROSPETTIVE FUTURE NEL RICONOSCIMENTO DI GESTURE E HUMAN ACTION RECOGNITION

per la classificazione di movimenti molto simili tra loro.

6.2 SFIDE APERTE E PROSPETTIVE FUTURE NEL RICONOSCIMENTO DI GESTURE E HUMAN ACTION RECOGNITION

Negli ultimi anni, ci sono stati notevoli progressi nel campo del riconoscimento di gesture e dell'human action recognition grazie alla crescita delle tecnologie di acquisizione dati e di elaborazione di immagini, degli algoritmi di apprendimento automatico e delle reti neurali convoluzionali.

Tuttavia, nonostante i progressi fatti, ci sono ancora diverse sfide aperte nel riconoscimento di gesture e dell'human action recognition.

Alcune delle sfide aperte nel riconoscimento di gesture e dell'human action recognition riguardano:

- **Variabilità delle gesture e delle azioni umane:** le gesture e le azioni umane possono variare in modo significativo in base a diversi fattori come l'ambiente, l'illuminazione, la postura del corpo, la velocità di esecuzione, il punto di vista della fotocamera e altri. Questa variabilità può rendere difficile il riconoscimento accurato delle gesture e delle azioni umane.
- **Dati di training limitati:** il riconoscimento di gesture e l'human action recognition richiedono grandi quantità di dati di training. Tuttavia, la raccolta di dati di training può essere costosa e richiedere molto tempo. Inoltre, i dati di training possono non essere rappresentativi di tutte le possibili situazioni in cui le gesture e le azioni umane possono essere eseguite.
- **Flessibilità limitata dell'algoritmo:** molti algoritmi di riconoscimento di gesture e di human action recognition richiedono una configurazione specifica e una taratura per funzionare correttamente. Ciò significa che l'algoritmo può non essere in grado di adattarsi facilmente a nuovi ambienti o a nuove situazioni.
- **Rumore e distorsioni nei dati di input:** i dati di input utilizzati possono essere soggetti a rumore e distorsioni. Ad esempio, le immagini acquisite possono contenere ombre, riflessi o rumore a causa di una bassa illuminazione o di un'acquisizione di bassa qualità. Questo rumore e queste distorsioni possono rendere difficile il riconoscimento accurato delle gesture e delle azioni umane.

Prospettive future nel riconoscimento di gesture e dell'human action recognition:

- **Deep learning:** l'apprendimento profondo è un'area di ricerca in rapida evoluzione nell'ambito dell'elaborazione di immagini e della visione artificiale. L'utilizzo di reti neurali convoluzionali profonde, ovvero reti neurali che prevedono un elevato numero di strati convoluzionali, può aiutare a superare le sfide del riconoscimento di gesture e dell'human action recognition.
- **Dati di training sintetici:** la creazione di dati di training sintetici può aiutare a superare il problema della limitatezza dei dati di training. Ciò può essere ottenuto utilizzando tecniche di sintesi di dati, come la realtà virtuale o la simulazione, che possono produrre dati di training che rappresentano una vasta gamma di situazioni e condizioni.
- **Sensori migliorati:** l'evoluzione delle tecnologie di sensori, come le telecamere ad alta risoluzione, i sensori di profondità e i sensori di movimento, può aiutare a migliorare la qualità dei dati di input e a superare il problema del rumore e delle distorsioni.
- **Approccio multimodale:** l'utilizzo di un approccio multimodale, che combina diverse modalità di input, come le immagini e i dati di sensori di movimento, può aiutare a migliorare l'accuratezza del riconoscimento di gesture e dell'human action recognition.

In conclusione, il riconoscimento di gesture e l'human action recognition sono temi importanti e in rapida evoluzione nell'ambito dell'elaborazione di immagini e della visione artificiale. Nonostante le sfide aperte esistenti, l'utilizzo di tecniche di deep learning, dati di training sintetici, sensori migliorati e approcci multimodali può portare a notevoli progressi nel riconoscimento accurato di gesture e azioni umane in diverse applicazioni.

Bibliografia

- [1] A.B. Sargano A. Mumtaz e Z. Habib. «Violence detection in surveillance videos with deep network using transfer learning». In: *2018 2nd European Conference on Electrical Engineering and Computer Science (EECS)*. IEEE. 2018.
- [2] T. Bouchrika A. Snoun e O. Jemai. «Deep-learning-based human activity recognition for Alzheimer's patients' daily life activities assistance». In: *Neural Computing and Applications* (2023).
- [3] J.K. Aggarwal e M.S. Ryoo. «Human Activity Analysis: A Review». In: *ACM Comput. Surv.* (2011).
- [4] O.C. Ann e L.B. Theng. «Human activity recognition: A review». In: *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*. 2014.
- [5] D.R. Beddiar. «Vision-based human activity recognition: a survey». In: *Multimed Tools Appl* (2020).
- [6] B. Bordel e R. Alcarria. «Recognizing human activities in Industry 4.0 scenarios through an analysis-modeling-recognition algorithm and context labels». In: *Integrated Computer-Aided Engineering* (2022).
- [7] X. Miao C. Yin J. Chen e H. Jiang. «Device-Free Human Activity Recognition with Low-Resolution Infrared Array Sensor Using Long Short-Term Memory Neural Network». In: *Sensors* (2021).
- [8] G. Zwoliński D. Kamińska e A. Laska-Leśniewicz. «Usability Testing of Virtual Reality Applications;The Pilot Study». In: *Sensors* (2022).
- [9] L. Chu F. Xiao L. Pei e D. Zou. «A deep learning method for complex human activity recognition using virtual wearable sensors». In: *Spatial Data and Intelligence: First International Conference, SpatialDI 2020, Virtual Event, May 8–9, 2020, Proceedings* 1. 2021.

BIBLIOGRAFIA

- [10] V.S.P. Patnam F.T. George e K. George. «Real-time deep learning based system to detect suspicious non-verbal gestures». In: *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE. 2018.
- [11] G. Chen H. Yan B. Hu e E. Zhengyuan. «Real-Time Continuous Human Rehabilitation Action Recognition using OpenPose and FCN». In: *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*.
- [12] S. Cang H. Yu e Y. Wang. «A review of sensor selection, sensor devices and sensor deployment for wearable sensor-based human activity recognition systems». In: *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*. 2016.
- [13] J. Yang H. Zou Y.Zhou e H. Jiang. «WiFi-enabled Device-free Gesture Recognition for Smart Home Automation». In: *2018 IEEE 14th International Conference on Control and Automation (ICCA)*. 2018.
- [14] K. Hos. «An overview of Human Action Recognition in sports based on Computer Vision». In: *Heliyon* (2022).
- [15] Y. Sun J. Chen e S. Sun. «Improving Human Activity Recognition Performance by Data Fusion and Feature Engineering». In: *Sensors* (2021).
- [16] M. Zaborski J. Gałka M. Maśior e K. Barczewska. «Inertial Motion Sensing Glove for Sign Language Gesture Acquisition and Recognition». In: *IEEE Sensors Journal* (2016).
- [17] R. Carotenuto L. Bibbò e F. Della Corte. «An Overview of Indoor Localization System for Human Activity Recognition (HAR) in Healthcare». In: *Sensor* (2022).
- [18] I.H. Lopez-Nava e A. Muñoz-Meléndez. «Human action recognition based on low- and high-level data from wearable inertial sensors.» In: *International Journal of Distributed Sensor Networks* (2019).
- [19] D. Baudry M. Dallel V. Havard e X. Savatier. «InHARD - Industrial Human Action Recognition Dataset in the Context of Industrial Collaborative Robotics». In: *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*. 2020.

- [20] D. Laurendi M. Merenda M. Astrologo e V. Romeo. «A Novel Fitness Tracker Using Edge Machine Learning». In: *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*. 2020.
- [21] Z. Ma e S.H. Ahmed. «Human Action Recognition in Smart Cultural Tourism Based on Fusion Techniques of Virtual Reality and SOM Neural Network». In: *Hindawi* (2021).
- [22] *MTw Awinda User Manual*. Xsens North America, Inc. 2018.
- [23] L.N.N. Nguyen e A. Cavallaro. «Basketball activity recognition using wearable inertial measurement units». In: *Proceedings of the XVI international conference on Human Computer Interaction*. 2015.
- [24] M. Abdurohman R.I. Fakhruddin e A.G. Putrada. «Improving PIR Sensor Network-Based Activity Recognition with PCA and KNN». In: *2021 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*. 2021.
- [25] A.S. Jalal R.K. Tripathi e S.C. Agrawal. «Suspicious human activity recognition: a review». In: *Artificial Intelligence Review* (2018).
- [26] B.H. Pawan Prasad G. S.R. Vishakha J. Akula e Rosh. «Posture Guided Human Action Recognition for Fitness Applications». In: Association for Computing Machinery, 2023.
- [27] M. Shehade e T. Stylianou-Lambert. «Virtual Reality in Museums: Exploring the Experiences of Museum Professionals». In: *Applied Sciences* (2020).
- [28] S. Ghanekar V. Rathod R. Katragadda e S. Raj. «Smart Surveillance and Real-Time Human Action Recognition Using OpenPose». In: Springer Singapore, 2020.
- [29] Y. Chen W. Zhuang e J. Su. «Design of human activity recognition algorithms based on a single wearable IMU sensor». In: *International Journal of Sensor Networks* (2019).

Ringraziamenti

Giunto al termine di questa esperienza formativa, desidero ringraziare di cuore tutti coloro che mi hanno accompagnato e guidato in questo importante traguardo.

In particolare, desidero esprimere il mio profondo ringraziamento al Prof. Stefano Ghidoni, relatore della mia tesi, per la sua preziosa guida e supporto durante tutto il percorso. Grazie anche a Matteo Terreran per la sua grande disponibilità e cortesia dimostratemi.

Desidero inoltre dedicare un sentito ringraziamento ai miei genitori, che mi hanno sostenuto con il loro amore, il loro supporto e il loro incoraggiamento costanti. A loro dedico questa tesi, frutto della dedizione e del sacrificio che mi hanno sempre trasmesso.

Non posso dimenticare di estendere i miei ringraziamenti a tutti i miei compagni di corso, che hanno arricchito la mia esperienza universitaria e mi hanno accompagnato in questa avventura. Un ringraziamento particolare va a Gianluca, compagno di molte esperienze condivise.

Infine, desidero ringraziare di cuore tutti gli amici e i familiari che mi hanno sostenuto durante l'intero percorso universitario.

Sono profondamente grato a tutti voi per aver reso questo percorso di studio e formazione un'esperienza indimenticabile. Grazie di cuore a ognuno di voi.

A

Appendice A: Codice in linguaggio C++ sviluppato per eseguire la conversione dei dati in formato utile per Edge Impulse

```
1 #include <ros/ros.h>
2 #include <rosbag/view.h>
3 #include <drapebot_msgs/SkeletonQuaternion.h>
4
5 #include <dirent.h>
6 #include <sys/stat.h>
7 #include <gtest/gtest.h>
8
9 #include <sstream>
10 #include <iostream>
11 #include <fstream>
12
13 #include <vector>
14 #include <string>
15
16 #include <boost/foreach.hpp>
17 #define foreach BOOST_FOREACH
18
19 //Struct used to store the data of one joint
20 struct Pose {
21     int segment_id;
22 }
```

```

23     double position_x;
24     double position_y;
25     double position_z;
26
27     double orientation_x;
28     double orientation_y;
29     double orientation_z;
30     double orientation_w;
31 };
32
33 int main ()
34 {
35     rosbag::Bag bag;
36     std::string path;
37     std::string directory_path;
38     std::string file_name;
39     std::vector<std::string> bag_files;
40     Pose pose;
41
42     std::cout << "Insert the path of the folder containing the files
43 to convert, starting from the /home directory" << std::endl;
44     std::cin >> directory_path;
45
46     // Pointer to directory
47     DIR *dir;
48
49     // Pointer to directory entry
50     struct dirent *ent;
51
52     // Structure for file status
53     struct stat st;
54
55     if ((dir = opendir(directory_path.c_str())) != NULL) {
56
57         // Read all files in the folder one at a timer
58         while ((ent = readdir(dir)) != NULL) {
59
60             // Create full file path
61             std::string file_path = directory_path + ent->d_name;
62
63             // Get file status
64             stat(file_path.c_str(), &st);

```

APPENDICE A. APPENDICE A: CODICE IN LINGUAGGIO C++ SVILUPPATO PER ESEGUIRE LA CONVERSIONE DEI DATI IN FORMATO UTILE PER EDGE IMPULSE

```
65     // Check if the entry is a regular file
66     if (S_ISREG(st.st_mode)) {
67
68         // Add file name to vector
69         bag_files.push_back(ent->d_name);
70     }
71 }
72
73 // Close the directory pointer
74 closedir(dir);
75 }
76 else {
77     std::cout << "Could not open directory" << std::endl;
78     return 1;
79 }
80
81
82
83 while(!bag_files.empty()){
84     std::string input;
85     std::stringstream ss(input);
86     std::string line;
87     std::string clean;
88     std::string temp;
89     std::string first_line = "timestamp,";
90
91     int timestamp = 0;
92     double start = 0;
93
94     file_name = bag_files.at(bag_files.size()-1);
95     bag_files.pop_back();
96
97     path = directory_path + file_name;
98     file_name = file_name.substr(0,13) + "csv";
99
100    std::cout << "Start conversion to: ";
101    std::cout << file_name << std::endl;
102
103    bag.open(path, rosbag::bagmode::Read);
104    rosbag::View view(bag);
105    std::ofstream file(file_name);
106
107    if(!file.is_open()){


```

```

108         std::cerr << "Error, file not open!" << std::endl;
109         return 1;
110     }
111
112     // Define first line of the .csv file containing fields name
113     for(int i = 1; i <= 23; i++ ){
114         first_line = first_line +
115             std::to_string(i) + "_position_x," +
116             std::to_string(i) + "_position_y," +
117             std::to_string(i) + "_position_z," +
118             std::to_string(i) + "_orientation_x," +
119             std::to_string(i) + "_orientation_y," +
120             std::to_string(i) + "_orientation_z," +
121             std::to_string(i) + "_orientation_w" ;
122
123         if(i != 23)
124             first_line = first_line + ",";
125     }
126
127     file << first_line << std::endl;
128
129     // Scans all the timestamp
130     foreach(rosbag::MessageInstance const m, view)
131     {
132
133         drapebot_msgs::SkeletonQuaternion::ConstPtr s = m.
134         instantiate<drapebot_msgs::SkeletonQuaternion>();
135
136         if (s != NULL){
137
138             // Save time in seconds to temp_time
139             double temp_time = s->header.stamp.sec +
140                 (s->header.stamp.nsec * 1e-9);
141
142             if (timestamp == 0) {
143                 // Save start time in seconds to start
144                 start = temp_time;
145             }
146
147             temp_time = temp_time - start;
148
149             // Time in milliseconds
150             temp = std::to_string((int)(temp_time*1000));

```

APPENDICE A. APPENDICE A: CODICE IN LINGUAGGIO C++ SVILUPPATO PER ESEGUIRE LA CONVERSIONE DEI DATI IN FORMATO UTILE PER EDGE IMPULSE

```
149
150     //Scans all the joint corresponding to a timestamp
151     for(int i = 0; i < s->skeleton.size(); i++){
152
153         ss << s->skeleton[i];
154
155         //Extract joint position and orientation data
156         while (getline(ss, line)) {
157             if (line.find("segment_id:") != std::string::
158                npos) {
159                 pose.segment_id = stoi(line.substr(line.
160                     find(":")+2));
161             }
162             if (line.find("position:") != std::string::
163                npos) {
164                 getline(ss, line);
165                 pose.position_x = std::stod(line.substr(
166                     line.find(":")+2));
167                 getline(ss, line);
168                 pose.position_y = std::stod(line.substr(
169                     line.find(":")+2));
170                 getline(ss, line);
171                 pose.position_z = std::stod(line.substr(
172                     line.find(":")+2));
173             }
174             if (line.find("orientation:") != std::string
175                 ::npos) {
176                 getline(ss, line);
177                 pose.orientation_x = std::stod(line.
178                     substr(line.find(":")+2));
179                 getline(ss, line);
180                 pose.orientation_y = std::stod(line.
181                     substr(line.find(":")+2));
182                 getline(ss, line);
183                 pose.orientation_z = std::stod(line.
184                     substr(line.find(":")+2));
185                 getline(ss, line);
186                 pose.orientation_w = std::stod(line.
187                     substr(line.find(":")+2));
188             }
189         }
190         temp = temp + "," +
191     }
```

```

181         std::to_string(pose.position_x) + "," +
182         std::to_string(pose.position_y) + "," +
183         std::to_string(pose.position_z) + "," +
184         std::to_string(pose.orientation_x) + "," +
185         std::to_string(pose.orientation_y) + "," +
186         std::to_string(pose.orientation_z) + "," +
187         std::to_string(pose.orientation_w);
188
189     }
190
191     file << temp << std::endl;
192     temp = "";
193     timestamp++;
194     ss.clear();
195
196 }
197 }
198
199 file.close();
200 bag.close();
201 }
202
203 return(0);
204
205 }
```