Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea Magistrale in Informatica

Implementazione e Valutazione di Architetture Deep Learning per la Predizione End-to-End della Qualità dei Processi Riabilitativi in Telemedicina

Relatore:
Chiar.mo Prof.
Stefano Pio Zingaro

Presentata da:
MICHELLE ZANOTTI

Sessione II
Anno Accademico 2022/2023

Alla mia anima,
alla mia metà.

A Matilde.

Abstract

Un processo riabilitativo - l'esecuzione di un determinato esercizio di riabilitazione - solitamente è analizzato e valutato da specialisti presenti durante lo svolgimento di questo , presso strutture mediche.
La telemedicina introduce la possibilità di valutare la qualità di un determinato processo riabilitativo da remoto sfruttando sistemi intelligenti ed esperti.
All'interno di questo lavoro di tesi ci si concentra sullo sviluppo di sistemi di Machine Learning, nel dettaglio sistemi Deep Learning supervisionati, in grado di fornire la qualità di un processo riabilitativo partendo dal video dell'esecuzione di un esercizio, con particolare attenzione rispetto alla possibile relazione presente tra la correttezza della predizione ottenuta e il livello di automazione introdotto da un modello.
Gli esercizi da valutare possono essere di cinque tipologie differenti e le valutazioni di questi sono composte da cinque aspetti distinti del movimento.
I video dello svolgimento degli esercizi subiscono alcune manipolazioni sfruttando la stima della posa di un individuo.
I framework sviluppati sono tre con automazione crescente, ma sfruttano le stesse architetture deep: Multi Layer Perceptron multi-output, per fornire la predizione della qualità, e Long Short Time Memory, per il riconoscimento dell'attività.
Si parte da un modello differente in base al tipo di esercizio da analizzare, successivamente viene introdotto il riconoscimento dell'attività al fine di poter utilizzare un unico modello indipendentemente dall'esercizio e infine viene implementata un'architettura con addestramento End-to-End.

ABSTRACT

Le performance dei modelli vanno di pari passo con l'aumento dell'automazione al loro interno, infatti l'architettura con il livello di automazione maggiore risulta essere anche quella con prestazioni migliori.
Il lavoro si conclude con una dimostrazione, tramite video non sfruttati per l'addestramento dei modelli, con la finalità di simulare il reale utilizzo di quanto implementato.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

La riabilitazione motoria è il processo essenziale per recuperare o potenziare le proprie funzionalità motorie a seguito di lesioni, interventi chirurgici, traumi, patologie o più in generale condizioni fisiologiche di una persona.

L'importanza di un percorso riabilitativo è sostenuta dal Ministero della Salute [1], il quale può essere definito come il terzo pilastro del sistema sanitario al fine di tutelare la cura del cittadino.
L'Organizzazione Mondiale della Sanità (WHO [2]) individua nella riabilitazione una delle chiavi sia per la copertura sanitaria universale sia per il raggiungimento del terzo *Sustainable Development Goal* – "Garantire una vita sana e promuovere il benessere per tutti a tutte le età"-.

Alcuni studi sulla riabilitazione stabiliscono che la gestione rapida delle conseguenze di eventi traumatici, ovviamente dal punto di vista motorio, è fondamentale per ottenere risultati soddisfacenti.
Quindi un programma di riabilitazione dovrebbe iniziare il prima possibile ed essere il più intenso possibile [6]. Questo non sempre è garantito con un approccio tradizionale.

[1] https://www.salute.gov.it/portale/lea/dettaglioContenutiLea.jsp?id=4720&area=Lea&menu=ospedaliera
[2] https://www.who.int/news-room/fact-sheets/detail/rehabilitation

Considerando un individuo che vorrebbe ripristinare la propria mobilità a seguito di un evento traumatico, solitamente l'intero processo di riabilitazione avviene presso strutture specializzate e sotto la supervisione di specialisti. La necessità di recarsi fisicamente in strutture sanitarie ostacola sia la rapidità con cui si intraprende un percorso di recupero fisico che la possibilità di accedervi. In dettaglio si ha una mancanza di accessibilità e barriere ai trasporti, soprattutto per coloro che vivono nelle aree rurali, i costi dei servizi e lunghi tempi di attesa [1] Per quanto riguarda le figure professionali, queste sono in grado di identificare le condizioni fisiche iniziali del paziente, definire un percorso di recupero motorio personalizzato, identificare le reali possibilità di recupero e di fornire una valutazione dell'intero processo.
Tutto ciò è possibile grazie alle loro competenze rispetto a movimenti, recupero e patologie.
Perciò, in una visione tradizionale di riabilitazione motoria, la presenza di questi specialisti è indispensabile al fine di ottenere il miglior trattamento possibile.
Esiste però un dislivello tra il numero di professionisti della riabilitazione disponibili e coloro che hanno necessità di intraprendere un percorso di riabilitazione.[1] Questo si traduce in tempi prolungati per fornire un riscontro rispetto alla qualità del processo riabilitativo svolto, andando a limitare l'intensità dell'intero processo.
D'altro canto però, tentando di ridurre queste tempistiche si possono presentare concessioni in termini di precisione nell'analisi.

Da queste considerazioni sorge una domanda: quale sarebbe l'impatto di un approccio più efficiente e accessibile?

È possibile ipotizzare un'alternativa innovativa alla riabilitazione tradizionale: la telemedicina basata sull'intelligenza artificiale, senza ausilio di dispositivi indossabili.
Questo sistema, in maniera analoga a un mentore virtuale, è in grado di valu-

tare immediatamente e in maniera accurata il progresso della riabilitazione. Questa soluzione elimina la necessità di spostamenti presso strutture cliniche garantendo valutazioni affidabili e qualitativamente confrontabili con quelle degli specialisti.

La telemedicina, definita dal Ministero della Salute[3] come il progresso della cura medica che attraverso la tecnologia permette la somministrazione di prestazioni mediche di vario genere da remoto, sta influenzando e rivoluzionando il panorama della riabilitazione e della medicina in generale.

La prima pubblicazione inerente alla telemedicina risale al 1950, dove viene descritta la trasmissione tramite telefono di immagini radiologiche; nei decenni successivi sono stati condotti diversi progetti focalizzati sulla telemedicina, per citarne alcuni: video comunicazioni, trasmissione di elettrocardiogrammi e comunicazione tramite satellite per fornire servizi medici agli astronauti [2].

Oggi vengono riconosciute come forme di telemedicina: visite, consulti, consulenze, assistenza, refertazioni di esami, monitoraggio dei parametri vitali effettuati a distanza con il supporto di computer, webcam, smartphone, sensori indossabili, connessione internet, e altro. Alcune applicazioni moderne della telemedicina hanno dato origine a dispositivi come SmartVista [3], bio sensori indossabili per l'assistenza remota dei pazienti cardiopatici, e di applicazioni mobile per il diabete scaricabili sul proprio smartphone che, integrando strumenti di documentazione dell'insulina basale, forniscono feedback automatizzati sui modelli glicemici degli utenti [4].

La telemedicina introduce alcuni vantaggi come:

- la riduzione di spostamenti e di dover assentarsi dal proprio posto di lavoro;

[3] https://www.salute.gov.it/portale/ehealth/dettaglioContenutiEHealth.jsp?lingua=italiano&id=5524&area=eHealth&menu=telemedicina

- l'aumento dell'accessibilità a cure mediche di alta qualità, soprattutto per coloro che vivono in zone rurali;
- la riduzione dei costi da affrontare sia per i pazienti che per le strutture sanitarie;
- la riduzione dei tempi di attesa;
- la rapida disponibilità di informazioni sullo stato di salute.

I vantaggi introdotti dalla telemedicina restano pressoché gli stessi anche se essa viene classificata in base al settore medico di applicazione.
Il progresso tecnologico introdotto con la telemedicina applicato ad un contesto di riabilitazione ha permesso la nascita della teleriabilitazione, una sottocategoria della telemedicina che fornisce un sistema di controllo della riabilitazione a distanza [5], rivoluzionando la cura e il monitoraggio delle persone in fase di riabilitazione.
La prima pubblicazione scientifica con oggetto la teleriabilitazione è datata 1998 ma negli ultimi anni il numero di articoli focalizzati su questo argomento è notevolmente aumentato [6].
Un approccio tele riabilitativo può fornire risultati positivi in diversi ambiti, neurologici [7], logopedici [8] o cardiologici [9], ma la sua principale applicazione avviene nella fisioterapia [6] e quindi in un contesto di riabilitazione motoria.

La teleriabilitazione per il ripristino della mobilità può avvenire con il supporto di dispositivi di assistenza robotica, esoscheletri, dispositivi tattili, ambienti di gioco virtuali oltre a sensori di motion capture a basso costo. [10]
Un'ulteriore possibilità è quella di utilizzare una semplice videocamera con il sostegno di elementi di Machine Learning e quindi Intelligenza Artificiale.

Infatti, grazie all'ausilio dei sistemi fondati sul Machine Learning [4], è og-

[4] https://www.ibm.com/topics/machine-learning

gi possibile effettuare valutazioni sulla qualità del processo riabilitativo in maniera rapidamente accessibile e decisamente affidabile, senza la costante presenza e supervisione di un professionista.
Il Machine Learning, quindi l'apprendimento automatico, è una sottocategoria dell'Intelligenza Artificiale[5] che permette di creare sistemi in grado di apprendere e migliorare le proprie performance in base ai dati che vengono utilizzati.
Entrando maggiormente nel dettaglio, il modo in cui un determinato modello apprende dai dati può essere supervisionato, utilizzando dati etichettati dai quali viene dedotta una relazione o funzione, oppure non supervisionato, dove la ricerca di pattern avviene utilizzando dati non etichettati.

L'utilizzo dell'intelligenza artificiale, grazie l'automazione che essa introduce, può garantire la rapidità necessaria nell'intraprendere un percorso di riabilitazione nonché assottigliare il divario presente tra specialisti della riabilitazione e quantità di pazienti che richiedono la loro assistenza.
Un sistema intelligente è in grado di acquisire dati, ad esempio video, inerenti all'esecuzione di un esercizio di riabilitazione, estrapolare le informazioni fondamentali e caratterizzanti di quanto è stato svolto e tramite quando ottenuto, come precedentemente detto, fornire una valutazione in modo rapido ed efficiente.
Ovviamente la presenza di un professionista, da remoto, è necessaria per le fasi che precedono l'esecuzione effettiva della riabilitazione.
Un aspetto fondamentale però è l'introduzione di sistemi basati sul Machine Learning facilmente utilizzabili senza possedere conoscenze tecniche e specifiche sul loro funzionamento, così che siano sfruttabili autonomamente da pazienti e specialisti.

Il connubio tra telemedicina e intelligenza artificiale in un ambito riabilitativo crea l'opportunità di intraprendere sessioni di riabilitazione nella

[5]https://www.oracle.com/it/artificial-intelligence/what-is-ai/

propria dimora, senza necessità di programmazioni continue di appuntamenti clinici, abbattendo così barriere di spazio e tempo [11].
Viene favorito l'accesso ad un determinato servizio a fasce di popolazione del tutto emarginate [11] e il monitoraggio in tempo reale dei progressi di un paziente.
Infine, l'unione di questi due mondi permette non solo di alleggerire gli oneri finanziari ma si rivela capace di migliorare la qualità delle cure, delineando uno scenario di cambiamento per chi si impegna nel processo di recupero.

Proposta di soluzione

All'interno di questo lavoro di tesi si vuole individuare la migliore configurazione di modelli di Machine Learning supervisionati per poter effettuare la previsione della qualitá del processo riabilitativo in telemedicina, concentrandosi sull' automazione del procedimento predittivo.

È importante sottolineare come con la dicitura automazione del procedimento predittivo non si vuole indicare il modo in cui un sistema produce una determinata previsione ma la riduzione della necessità dell'intervento umano per far si che un determinato framework, costituito da diversi step, possa concludere il proprio compito.

Il punto di partenza di questa ricerca è individuabile in video raffiguranti sessioni di diversi esercizi di riabilitazione svolti da pazienti, con differente età e condizione fisica, forniti dall'Istituto Ortopedico Rizzoli[6] di Bologna. Lo stesso istituto, ha inoltre fornito una specifica modalità di valutazione del processo riabilitativo.
Queste condizioni hanno contribuito allo sviluppo di modelli ad hoc, perciò approcci che, partendo da quanto descritto in letteratura, sono stati adattati

[6]Istituto Ortopedico Rizzoli, Via di Barbiano, 1/13, 40136 Bologna, Italia

al contesto e al materiale a disposizione.

In particolare, sono stati presi in considerazione elementi riguardanti:

- l'ambito riabilitativo;
- l'analisi di video;
- activity recognition e pose estimation;
- action quality assessment.

Un aspetto molto importante riguarda lo sviluppo e l'implementazione di tre diversi framework, dal punto di vista della conformazione, ma simili rispetto a tecniche e tecnologie utilizzate, al fine di individuare la soluzione più adatta.
Le tecnologie in questione risultano essere: Multi Layer Perceptron multi-output, per fornire la predizione della qualità, e Long Short Time Memory, per il riconoscimento dell'attività.

La differenza sostanziale risulta essere nell'automazione del processo predittivo, ponendo una domanda fondamentale: **una maggior automazione**, quindi una riduzione dell'intervento umano durante la fase predittiva, **può tradursi in una maggior precisione della previsione ottenuta?**

Quanto prodotto all'interno di questo lavoro di tesi risulta avera una rilevanza accademica e pratica sia nell'ambito informatico e intelligenza artificiale che medico.
Dal punto di vista medico può essere visto come lo step iniziale per intraprendere un servizio di teleriabilitazione con buoni risultati e costi limitati.
Dal punto di vista informatico, la necessità di creare una soluzione con vincoli rigidi ha permesso la nascita di framework particolari, sfruttando l'unione di tecnologie non sempre utilizzate in modo congiunto e probabilmente utilizzabili anche in contesti differenti.

Nei prossimi capitoli verrà mostrato con maggior precisione e attenzione il processo di sviluppo partito con la creazione di modelli specifici per ogni esercizio riabilitativo svolto e concluso con l'implementazione di un'architettura End-to-End, dove l'unico intervento umano è individuabile nell'avvio del processo predittivo.
Nel capitolo 2 verrà riportata la revisione della letteratura e in particolare i lavori che propongono soluzioni in parte simili a quanto è stato sviluppato.
Nel capitolo 3 verrà effettuata una descrizione dei metodi e dei materiali utilizzati per la sperimentazione. Verrà esplicitata la metodologia e modelli utilizzati, oltre a come sia stata eseguita l'acquisizione delle informazioni salienti partendo dai dati a disposizione. Inoltre verrà mostrato il setup della sperimentazione eseguita.
Nel capitolo 4 verranno visionati e discussi i risultati ottenuti, con particolare attenzione sulla relazione tra automazione introdotta in ogni modello e i risultati ottenuti. In aggiunta verrà visionata una demo sviluppata sfruttando nuovi dati.
Infine verranno presentate le conclusioni e le possibili direzioni future.

Capitolo 2

Revisione della Letteratura

In questo capitolo verrà fatta una revisione dei lavori presenti in letteratura inerenti a due contesti collegati a questa tesi:

- **ambito riabilitativo**;
- **activity recognition** e l'utilizzo della pose estimation per riconoscere un'attività svolta all'interno di un video.

Ambito Riabilitativo

Con ambito riabilitativo si vuole indicare lo sviluppo di framework e modelli con la finalità di monitorare, valutare e correggere lo svolgimento di esercizi di riabilitazione fisica partendo da video.

All'interno di [12] viene presentato un framework basato sull'apprendimento automatico che identifica gli errori commessi da un utente durante lo svolgimento di un esercizio e propone misure correttive individuali e personalizzate.
Per fare ciò viene utilizzata una rete neurale caratterizzata da due rami: un classificatore per gli errori commessi e un correttore per fornire le misure correttiva. In entrambi i casi viene fatto affidamento su una Graph Convolutional Network.

Quanto sviluppato sfrutta gli ambiti della pose estimation, dell'activity recognition e ovviamente della previsione del movimento.

In [13] viene proposto un framework che sfrutta metodi di deep learning supervisionato per la valutazione automatizzata della qualità degli esercizi di riabilitazione fisica.
Vengono usate metriche, come la Log-Likelihood, per quantificare le prestazioni del movimento e introdotte funzioni di scoring per mappare i valori delle metriche delle prestazioni di movimento in punteggi di qualità nell'intervallo [0,1].
Il documento, inoltre, introduce una rete neurale che sfrutta le caratteristiche spaziali dei movimenti umano tramite un'elaborazione gerarchica degli spostamenti delle diverse parti del corpo.
In questo caso i dati utilizzati per la convalida sono stati acquisiti con il sistema di tracciamento ottico Vicon (117 sequenze dimensionali di angular joint displacements).

Il lavoro effettuato e descritto in [14] vuole dimostrare se sia possibile, sfruttando metodi di apprendimento automatico, eseguire una classificazione binaria sulla correttezza dell'esecuzione di azioni relative a esercizi di riabilitazione fisica.
Viene sfruttata una rete neurale convoluzionale, Res-TCN, e metodi standard come Support Vector Machine e Random Forest.
I dati utilizzati per questo studio sono stati acquisiti sia tramite fotocamera Kinect che sistema di tracciamento ottico Vicon e consistono in dieci movimenti svolti da dieci individui ripetuti dieci volte.
Vengono utilizzati sia la posizione di punti 3D che angoli di congiunzione.

Nell'articolo [15] viene presentata una nuova tecnologia basata su computer vision e machine learning per il monitoraggio delle articolazioni del corpo di pazienti durante fisioterapia riabilitativa. Vengono presi in considerazione

esercizi come squat, estensione dell'anca e flessione del ginocchio.
Viene introdotta un'architettura di rete neurale composta da due moduli: uno per la rilevazione degli esercizi e uno per la misurazione della correttezza di tali esercizi.
Il processo che caratterizza questa soluzione parte con l'acquisizione video dell'esecuzione di un esercizio passando poi all'identificazione delle articolazioni tramite la libreria OpenPose; successivamente avviene la validazione dell'esercizio sfruttando il modulo di misurazione dell'architettura ed infine vengono rappresentanti a video i risultati ottenuti.

Tutte le soluzioni trovate in letteratura hanno fornito un diverso approccio e punto di vista rispetto alla predizione della qualità del movimento in processi riabilitativi, con particolare attenzione all'automazione di questo.
Quanto studiato è stato un punto di partenza per questo lavoro di tesi, il quale però, avendo vincoli dovuti a video e al calcolo delle qualità del movimento, ha richiesto un approccio differente rispetto a quanto precedentemente implementato e descritto in letteratura.

Activity Recognition

Con action recognition si vuole indicare l'ampio campo di studio che si focalizza sul riconoscimento di un'azione umana da un video contenente l'esecuzione completa dell'azione stessa.

Tutti gli articoli presi in considerazione [16, 17, 18] hanno tre caratteristiche fondamentali in comune, e risultate fondamentali per questo lavoro di tesi.
In primo luogo in tutti e tre i lavori vengono utilizzati video RGB, senza sfruttare particolari tecnologie o sensori indossabili.
In secondo luogo viene effettuata l'estrazione dei dati dello scheletro da video, quindi di punti chiave della posa e del movimento svolto all'interno di

un video, tramite framework per pose estimation.
Infine il modello scelto per il riconoscimento dell'azione svolta è il Long Short Term Memory (LSTM), una rete neurale ricorrente che conserva informazioni salienti di sequenze temporali.

Capitolo 3

Metodi e Materiali

All'interno di questo capitolo verrà presentato il dataset utilizzato in questo lavoro di tesi, video e panel, con alcune considerazioni sulla distribuzione dei dati al suo interno.
Successivamente verranno descritte le due librerie Python risultate fondamentali per la manipolazione dei video.
In seguito verranno introdotte le due principali architetture utilizzare per ottenere il riconoscimento di un'attività partendo da un video e la valutazione di tale attività.
Il capitolo si conclude con il setup della sperimentazione, ponendo particolare attenzione verso librerie e tecnologie sfruttate per la creazione di tre framework, il processo che ha condotto all'implementazione di questi e le loro caratteristiche fondamentali.

3.1 Dataset

Il dati utilizzati all'interno di questo lavoro sono stati forniti dall'Istituto Ortopedico Rizzoli di Bologna.
L'Istituto ha reso disponibili sia video di pazienti durante lo svolgimento di esercizi di riabilitazione, sia panel compilati da medici con le valutazioni rispetto all'esecuzione di un esercizio.

Sono stati utilizzati, però, solamente video che avessero il corrispettivo panel e viceversa.

Video

I video, in formato MP4, rappresentano l'esecuzione di 5 differenti esercizi riabilitativi, focalizzati sulla parte superiore del corpo:

1. elevazione anteriore;
2. abduzione;
3. extrarotazione;
4. intrarotazione;
5. elevazione e rotazione superiore.

e sono organizzati in cartelle rispetto all'esercizio svolto.

Per avere una migliore comprensione della composizione e distribuzione dei dati a disposizione, è stata fatta un'analisi visualizzando i dati tramite Matplotlib[1], conteggiando il numero di video forniti rispetto al tipo di esercizio eseguito al loro interno.
Il risultato è presente nella figura 3.1.

All'interno dell'immagine è possibile notare come l'esercizio con maggior elementi sia l'extrarotazione e quello con il minor numero l'abduzione.
Ovviamente, è necessario sottolineare che non esiste una netta differenza tra la quantità dei vari esercizi perciò ci si trova davanti ad un dataset bilanciato.

[1] https://matplotlib.org/

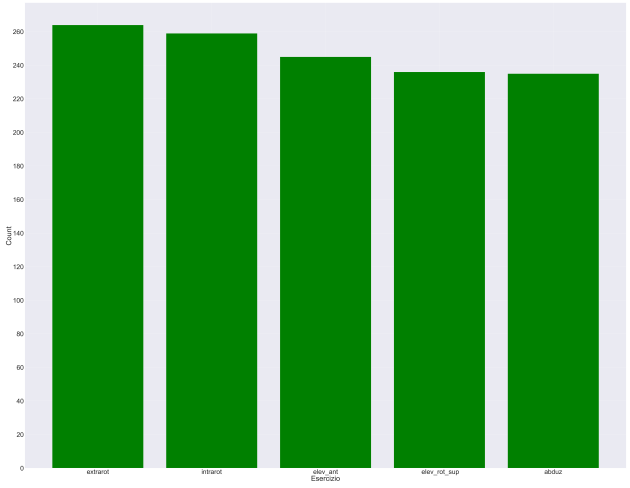

Figura 3.1: Distribuzione nel dataset delle tipologia di esercizio

I video non sono stati utilizzati nel formato nel quale sono stati forniti, ma sono stati processati, al fine di ottenere sequenze temporali delle coordinate (x,y) di alcune parti del corpo, tramite MediaPipe[20] e OpenCV [19]. Le tecnologie menzionate e il processo seguito verrà descritto in modo esaustivo all'interno della sezione 3.2.

Panel

I panel sono file in formato Excel compilati da medici specializzati e rappresentano la loro valutazione rispetto all'esecuzione di un determinato esercizio. La modalità di valutazione del processo riabilitativo risulta essere complessa poiché composta da più elementi e non semplicemente da un punteggio finale. Per ogni esercizio vengono presi in considerazione diversi aspetti del movimento eseguito:

- obiettivo (è stato raggiunto l'obiettivo primario dell'esercizio?);
- ampiezza (è completa l'ampiezza del movimento?);
- capo (è corretta la postura del capo?);
- spalla (è corretta la postura della spalla?);

- tronco (è corretta la postura del tronco?).

Il punteggio attribuito ad ogni caratteristica rientra nel range [0,5] e la somma di questi permette di ottenere il cosiddetto *Score*, il quale, ha come valore massimo 25.
Un esempio di panel fornito è presente nella immagine sottostante.

Punteggi	1= Per nulla	2= Poco	3= Abbastanza	4= Molto	5= Moltissimo		1.ELEV ANT
REP SES 1	Obiettivo	Ampiezza	Capo	Spalla	Tronco	Score	E' stato raggiunto l'obbiettivo primario dell'esercizio?
1	5	4	5	4	4	22	E' completa l'ampiezza del movimento?
2	5	4	5	4	4	22	E' corretta la postura del capo?
3	5	4	5	4	4	22	E' corretta la postura della spalla?
4	5	4	5	4	4	22	E' corretta la postura del tronco?
5	5	4	4	4	4	21	
6	5	4	5	4	4	22	
7	5	4	5	4	4	22	
8	5	4	5	4	4	22	
9	5	4	5	4	4	22	
10	5	4	4	4	4	21	
						218	

Figura 3.2: Esempio di panel

Come è possibile notare nella figura 3.2, i dati forniti risultano essere visivamente chiari.
Nonostante le caratteristiche estetiche, però, la struttura e tipologia dei file risulta essere poco adeguata per un'analisi automatica.
Perciò, per semplificare il loro utilizzo, sono stati esportati e salvati in formato CSV (figura 3.3) fornendo un conformazione tabellare facilmente acquisibile e manipolabile.

Obiettivo	Ampiezza	Capo	Spalla	Tronco	Score
5	4	5	4	4	22
5	4	5	4	4	22
5	4	5	4	4	22
5	4	5	4	4	22
5	4	4	4	4	21
5	4	5	4	4	22
5	4	5	4	4	22
5	4	5	4	4	22
5	4	5	4	4	22
5	4	4	4	4	21

Figura 3.3: Esempio di panel in formato CSV

Anche nel caso dei panel, così come per i video, è stata fatta un'analisi della distribuzione dei valori dei vari aspetti del movimento da considerare rispetto ai cinque esercizi forniti. Per fare ciò vengono sfruttate le librerie

python Matplotlib[2] e seaborn[3].
Sono stati creati cinque *countplot*[4], i quali permettono di confrontare i valori assunti da un determinato aspetto del movimento nei vari esercizi.
I grafici sono descritti di seguito.

In 3.4 è presente la distribuzione di *Obiettivo*. Si può notare come per l'abduzione e l'elevazione e rotazione superiore non sono presenti valori uguali a 1, mentre per l'intrarotazione non sono presenti valori uguali a 5.

Il grafico 3.5 rappresenta la distribuzione di *Ampiezza*. In questo caso mancano valori uguali a 5 e 1 rispettivamente per l'intrarotazione e l'elevazione e rotazione superiore, la quale inoltre, ha solamente due unità con valore 5.

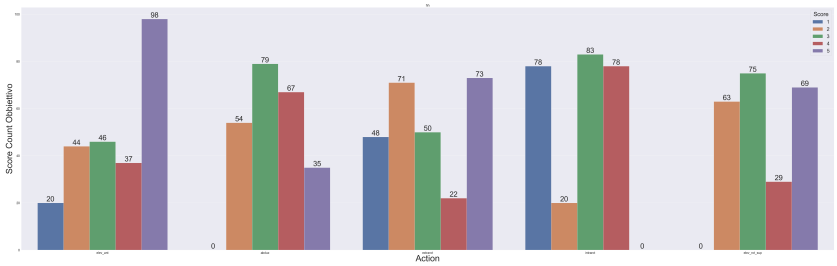

Figura 3.4: Distribuzione valori di *Obiettivo* per esercizio

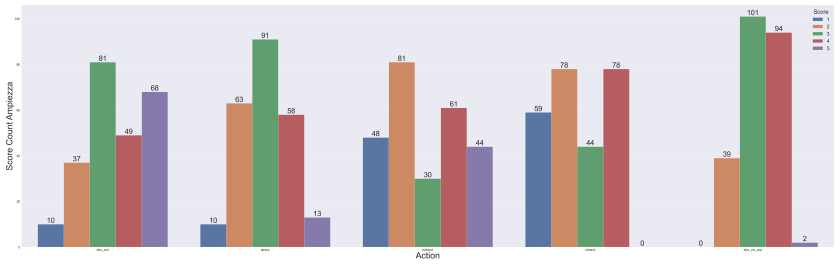

Figura 3.5: Distribuzione valori di *Ampiezza* per esercizio

[2]https://matplotlib.org/
[3]http://seaborn.pydata.org/index.html
[4]http://seaborn.pydata.org/generated/seaborn.countplot.html

La rappresentazione della distribuzione di *Capo* è nella figura 3.6. Per questo aspetto del movimento mancano, per ogni tipo di esercizio, valori pari a 1 e si hanno principalmente valutazioni uguali a 5.

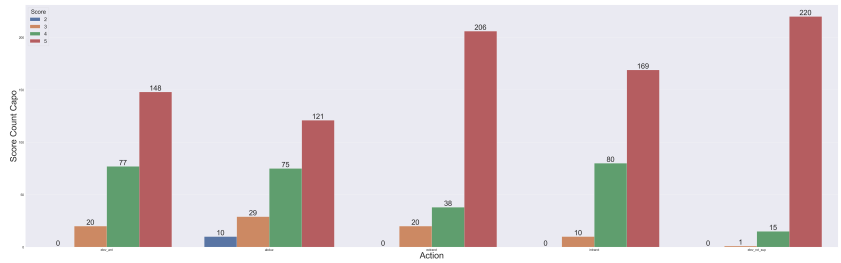

Figura 3.6: Distribuzione valori di *Capo* per esercizio

Spalla e *Tronco*, la cui rappresentazione è presente in 3.7 e 3.8, hanno una distribuzione molto simile: mancanza di valori uguali a 1 per ogni tipo di esercizio, presenza quasi nulla di valori pari a 2 e per l'elevazione e rotazione superiore nessun valutazione con valore di 2, 3, 4.

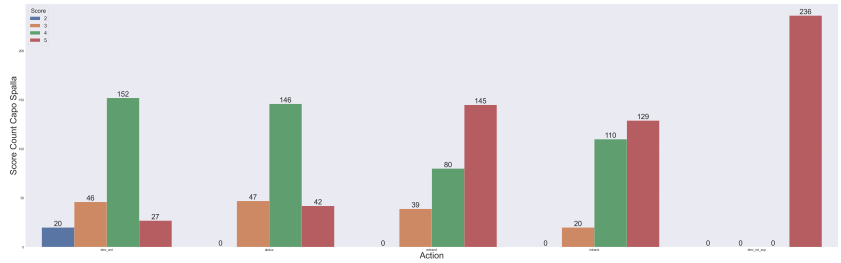

Figura 3.7: Distribuzione valori di *Spalla* per esercizio

Globalmente è possibile evidenziare la totale assenza di valutazioni uguali a 0, per ogni aspetto in ogni esercizio.
La presenza molto spesso di valutazioni alte (4-5) è da attribuire al tipo di esercizio a cui ci si sta riferendo.

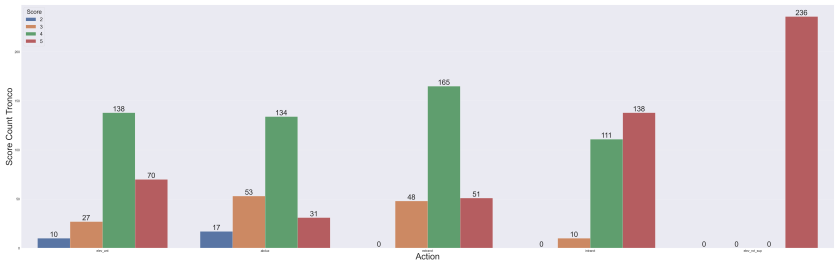

Figura 3.8: Distribuzione valori di *Tronco* per esercizio

3.2 OpenCV e MediaPipe

MediaPipe [20] è un framework open source cross platform per la creazione di pipeline di machine learning con lo scopo di processare serie temporali, come video e audio. Il toolkit messo a disposizione da MediaPipe comprende:

- MediaPipe Framework, componente di basso livello utilizzato per creare pipeline efficienti on-device. [20]
- MediaPipe Solution, insieme di librerie e strumenti che permettono di applicare tecniche di intelligenza artificiale e machine learning in un'applicazione. [20]

All'interno di solution troviamo diversi task, tra i quali *Pose landmark detection* che permette di identificare i punti chiave di un corpo umano all'interno di immagini e video.

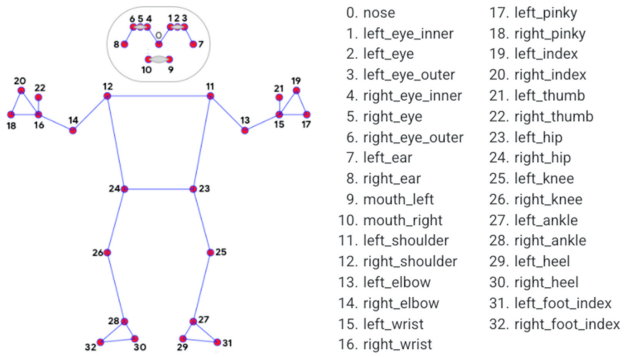

Figura 3.9: MediaPipe pose landmark detection

Rispetto ai 33 punti di riferimento del corpo umano messi a disposizione dal task di MediaPipe, ai fini di questo lavoro di tesi, ne vengono presi in considerazione solamente 9:

- naso;
- spalla destra e sinistra;
- gomito destro e sinistro;
- polso destro e sinistro;
- anca destra e sinistra.

Figura 3.10: Punti di riferimento sfruttati

Per ognuno di questi 9 punti, visibili nell'immagine 3.10, vengono considerate le rispettive coordinate 2D (x,y) per ogni frame del video che si sta analizzando.

OpenCV [19], invece, è una libreria di computer vision e machine learning con più di 2500 algoritmi *classici* e *state-of-art*.
Ovviamente il suo utilizzo è nell'ambito della computer vision.
Questa libreria viene sfruttata in collaborazione con il rilevamento dei punti chiave di un corpo.
Il contributo apportato risulta essere: la possibilità di caricare un video,

settare gli fps, modificare il colore del video che si sta analizzando e disegnare i punti chiave forniti da MediaPipe.

3.3 LSTM

L'architettura long short-term memory LSTM è stata proposta nell'articolo [22] ed è diventata, grazie alla sua applicazione in molti ambiti, un modello definito state-of-the-art.
Risulta essere una rete neurale ricorrente che conserva informazioni salienti di sequenze temporali.
Questo modello è stato creato per eliminare il problema del exploding/vanishing gradient, il quale può verificarsi in altre tipologie di reti neurali ricorrenti. [23]

All'interno di questo lavoro è stato utilizzato il modello definito *Vanilla*. LSTM Vanilla partendo dalla prima architettura proposta incorpora cambiamenti proposti in [24] e [25].

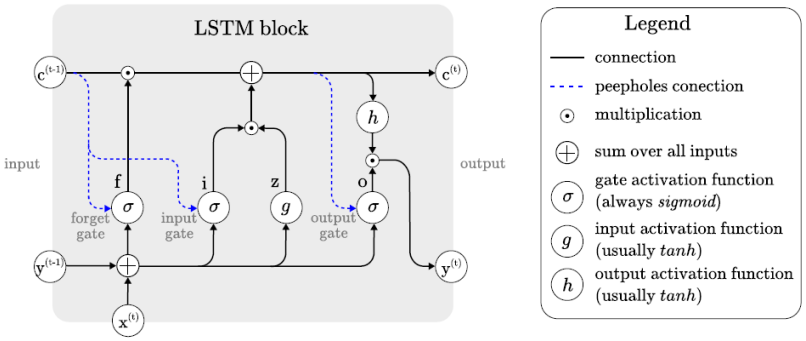

Figura 3.11: architettura di un blocco LSTM vanilla [26]

Nella figura 3.11 viene presentata l'architettura di un blocco LSTM vanilla. Il processo di forward può essere suddiviso in cinque step [26, 21]:

1. Block input, aggiornamento del block input combinando l'input corrente e l'output dell'ultima iterazione;

2. Input gate, aggiornamento dell'input gate combinando l'input corrente, l'input e il valore c (cell value) dell'ultima iterazione;

3. Forget gates, l'unità LSTM decide quale informazione deve essere rimossa dalle sue cell states precedenti;

4. Cell, calcolo di c che combina i valori del block input, dell'input gate e del forget gate con il valore precedente di c;

5. Output gates, calcolo dell'output combinando l'input corrente, l'output dell'unità LSTM e il valore c dell'ultima iterazione.

Esiste la possibilità di utilizzare più layer LSTM.

La variante vanilla, però, all'interno di questa tesi, viene utilizzato in modo bidirezionale, parlando perciò di bi-directional long short-term memory (Bi-LSTM).
La particolarità di un LSTM bidirezionale risulta essere il fatto che siano presenti, per ogni layer, due reti: una per eseguire i passi di forward accedendo alle informazioni passate e una per eseguire gli step di backward accedendo alle informazioni successive. Entrambe le reti sono però connesse allo stesso layer di output.

Perchè LSTM?

La scelta di utilizzare un'architettura di tipo LSTM è stata influenzata dallo scopo da raggiungere - il riconoscimento dell'attività svolta all'interno di un video - e dall'input utilizzato - sequenze temporali di coordinate che rappresentano lo spostamento delle articolazioni di un individuo.-
LSTM, infatti, risulta essere ampiamente utilizzato in contesti di questo tipo (activity recognition skeleton-based). [27]

Perchè Bidirezionale?

Anche l'utilizzo di una versione bidirezionale di LSTM è collegato a quanto menzionato prima: il riconoscimento di un'azione necessità la conoscenza, date le coordinate ad un tempo t, di quali fossero le coordinate a t-1 e quali saranno quelle a t+1.
Bi-LSTM opera esattamente in questo modo.

3.4 MLP

Un multi layer perceptron, o MLP, è una variante del modello di Perceptron proposto in [28] nel 1958.

L'architettura di questa tipologia di rete profonda è caratterizzata da uno o più layer hidden situati tra le informazioni in input (input layer) e l'output restituito (output layer), il cui output non viene esplicitamente mostrato.
I neuroni che compongono ogni layer vengono definiti *full-connected* poiché sono collegati ad ogni neurone del layer precedente, da cui ricevono informazioni, e ad ogni neurone del layer successivo, a cui passano informazioni.

L'addestramento di reti di questo tipo avviene sfruttando: discesa del gradiente (o una delle sue varianti), passi di forward e passi di backward.

Discesa del gradiente sfrutta il calcolo delle derivate per poter modificare pesi e bias di ogni layer della rete al fine di minimizzare la funzione di costo, errore tra ciò che è stato calcolato dalla rete e il valore reale;

Passi di forward [21] propagano le informazioni ricevute dagli input attraverso le unità di ogni hidden layer per produrre l'output;

Passi di backward [21] consentono a quanto appreso ,calcolando la funzione di costo, di ripercorrere a ritroso la rete permettendo il calcolo del gradiente.

Un multi layer perceptron può essere utilizzato per risolvere problemi sia di regressione che di classificazione e, all'interno di questo lavoro di tesi, sono stati presi in considerazione e sviluppati entrambi i casi.

In particolare, però, decidendo di affrontare la previsione della qualità del movimento come un problema di classificazione e avendo una definizione specifica della previsione, come spiegato in 3.1, il MLP utilizzato è di tipo multi-output.
Con la dicitura multi-output si vuole indicare il fatto che all'interno del layer di output è presente più di un'unità ma ogni unità potrà assumere uno ed un solo valore tra quelli previsti.

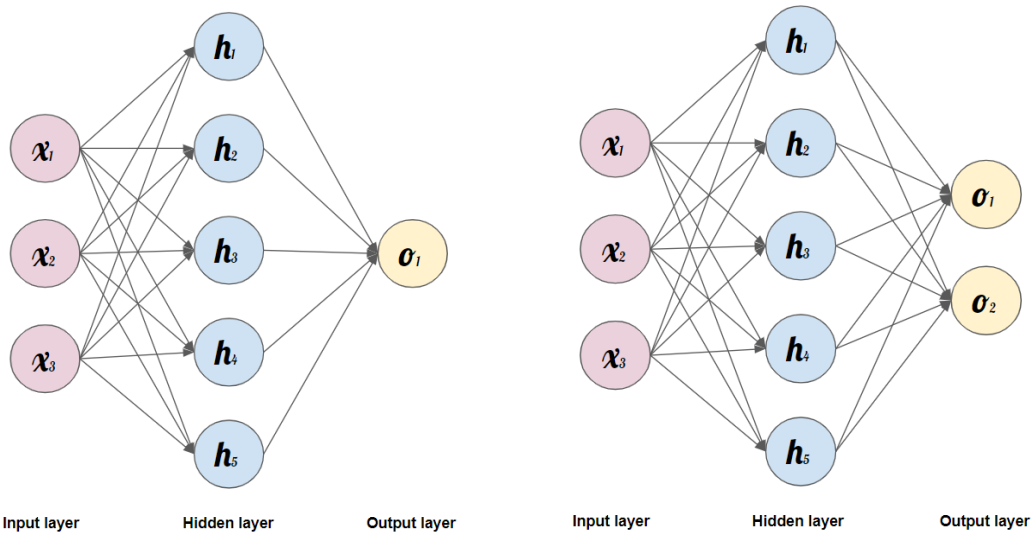

Figura 3.12: MLP e Multi-output MLP

3.5 Setup della Sperimentazione

Il linguaggio di programmazione utilizzato per lo sviluppo di questo lavoro di tesi è Python nella versione 3.9.0.
Python risulta essere uno dei linguaggi di riferimento in ambito Machine Learning e Intelligenza Artificiale, questo perché mette a disposizione librerie e framework che permettono di sfruttare modelli di ML tradizionali non-deep (ex. SVM, random forest, ecc..) e deep, già implementati senza la necessità di crearli ex novo.
Inoltre, in generale, è un linguaggio di programmazione che si presta molto bene per gestione/manipolazione/analisi dei dati e calcolo scientifico.

Innanzitutto è stata sfruttata Pytorch [29], una libreria basata sul framework Torch sviluppato principalmente da Meta AI (laboratorio di intelligenza artificiale appartenente alla società Meta), nella versione 2.0.1 con il supporto della piattaforma CUDA versione 11.7.0.
Pytorch può essere visto come un sostituto della libreria NumPy in grado di utilizzare la potenza di calcolo delle GPU e di altri acceleratori e ovviamente utilizzata per la creazione di modelli di Deep Learning, nel caso di questo lavoro di tesi MLP e LSTM.
Lo sviluppo di un modello di Deep Learning sfruttando questa libreria è facilitato dai moduli messi a disposizione da essa come ad esempio una vasta gamma di ottimizzatori e scheduler, sistemi di autograd per il calcolo dei gradienti e di backpropagation.
Quindi Pytorch è stato utilizzato per la creazione di architetture di reti neurali, l'addestramento e la validazione dei modelli creati. Inoltre, per poter acquisire i dati salvati in file CSV, la libreria è stata sfruttata per la creazione di oggetti Dataset e successivamente DataLoader.

Prendendo in considerazione i dati a disposizione, la manipolazione dei video è avvenuta tramite le librerie MediaPipe (versione 4.7.0.72) e OpenCV (versione 0.9.2.1), presentate in 3.2.

Invece, per quanto riguarda i panel sono state utilizzate librerie come Pandas (versione 1.5.3) e NumPy (versione 1.23.5) per la loro trasformazione in formato CSV.

Per questo lavoro di tesi, avente la finalità di capire se una maggior automazione del processo predittivo corrisponda ad una previsione più accurata, è stato necessario sviluppare diversi framework con ovviamente diversi livelli di automazione:

1. creazione di un classificatore tramite MLP multi-output per ogni tipologia di esercizio (basso livello);
2. creazione di un unico classificatore sfruttando il riconoscimento dell'attività tramite LSTM, avendo però due architetture separate (medio livello);
3. creazione di un'architettura end-to-end valutando la possibilità di passare da un classificatore multi output alla regressione e modificando il modo in cui le informazioni estratte da lstm vengono usate nella creazione della classificazione finale.

Nonostante il diverso tipo di automazione che caratterizza ogni framework esistono alcuni elementi in comune.
In particolare, per quanto riguarda l'addestramento dei vari modelli si è deciso di utilizzare la CrossEntropyLoss (nel caso di classificatore), la MSELoss (nel caso di regressione), come ottimizzare lo Stochastic Gradient Descent (SGD) e come scheduler dell'ottimizzatore lo ExponentialLR.

La CrossEntropyLoss [30] è una funzione costo utilizzata principalmente in situazioni in cui ci si trova in un problema di classificazione.
Questa funzione si basa sulla teoria dell'entropia e calcola la differenza della distribuzione di probabilità dati, in questo caso, gli score presenti nei panel e gli score calcolati dal modello o l'attività riconosciuta.

Ciò che accade durante l'addestramento di un modello risulta essere la minimizzazione di questa funzione andando quindi a ridurre la differenza delle distribuzioni di probabilità e quindi a massimizzare le capacità di previsione del modello stesso.
Ovviamente, avendo considerato la possibilità di ottenere la predizione della qualità del processo riabilitativo attraverso un approccio di regressione è stato necessario sostituire la funzione di costo appena descritta con una funzione adatta a problemi di regressione, la MSELoss [31].
Anche in questo caso, ciò che avviene durante l'addestramento è la minimizzazione della loss ma calcolando l'errore quadratico medio tra un input x e un target y.

Per poter minimizzare la funzione costo, qualunque essa sia, è necessario utilizzare un ottimizzatore che permetta la modifica di bias e pesi della rete. L'ottimizzatore scelto è il Stochastic Gradient Descent [32].
Questa è un'alternativa stocastica della discesa del gradiente tradizionale, citata in 3.4, che calcola le derivate solamente su un sottoinsieme del set di training. Utilizzando i cosiddetti *mini-batch*, questo tipo di ottimizzare rende l'addestramento più veloce.
Al SGD è stato affiancato uno scheduler, ExponentialLR [33], che permettesse il decadimento della velocità di apprendimento di ciascun gruppo di parametri in modo esponenziale.

Oltre a quanto menzionato, la fase di assestramento prevede *Early Stop*, una forma di regolarizzazione, con una *patience* di cinque.
Utilizzare questa tecnica significa che durante il training del modello si considera l'errore calcolato sulla parte di dataset utilizzata per la validazione, nel caso in cui questo continui ad aumentare per cinque epoche consecutive l'addestramento terminerà.

Nei framework che prevedono l'utilizzo di unità LSTM, l'input ricevuto

subisce *batch normalization*. [34]
Ciò che accade è la normalizzazione degli elementi contenuti in un batch portandoli a valori tra 0 e 1.
Questo è utile poiché le coordinate, fornite all'architettura per il riconoscimento dell'attività, hanno una varianza spesso alta e la riduzione dei loro valori rispetto a un range definito potrebbe portare conseguenze positive all'addestramento del modello.

Infine sia per il riconoscimento dell'attività che per il calcolo della qualità del processo riabilitativo il dataset viene suddiviso in train, validation e test set sfruttando la libreria Scikit Learn [35] e il suo modulo train_test_split [36]. In particolare abbiamo una suddivisione iniziale del dataset in train, 80%, e test, 20%, e successivamente il train set viene suddiviso tra train e validation set dove al validation viene attribuita il 20% del train set precedentemente calcolato.

L'utilizzo di tecniche deep, quindi di reti neurali, è stato preceduto, però, da una fase sperimentale che utilizza tecniche supervisionate non-deep: SVM, random forest, KNN, alberi decisionali.
Nonostante la totale assenza di automazione in questa fase di sviluppo, essa ha fornito un contributo essenziale a ciò che è stato sviluppato successivamente.
I vari framework verranno dettagliati in seguito.

Tecniche Supervisionate non-deep

Le tecniche di ML supervisionate tradizionali e il loro utilizzo, sono state il punto di partenza per ciò che è stato sviluppato in questo lavoro di tesi.
Ovviamente il dataset di partenza è lo stesso, quindi panel e video.
Durante questa fase di sviluppo, che può essere definita primordiale, l'automazione è totalmente assente ma ha permesso di comprendere quale fosse la

direzione da seguire.
Il contributo apportato a questo lavoro di tesi può essere così riassunto:

- modifica del formato dei panel da excel a CSV;
- utilizzo delle librerie MediaPipe e OpenCV per l'estrazione dei punti chiave del corpo umano da video e per la loro successiva manipolazione. I punti chiave saranno differenti per ogni esercizio.

Entrando nel dettaglio della manipolazione di questi punti chiave, viene eseguita la media dei valori corrispondenti a tre frame (primo, medio, ultimo) per ogni coordinata (x,y) di ogni punto chiave considerato.
I punti forniti da MediaPipe non comprendono esplicitamente informazioni riguardanti il tronco, il quale però è un aspetto fondamentale da considerare nella predizione della qualità del movimento in questo caso.
Ciò che è stato fatto è calcolare la postura del tronco tramite la differenza tra il punto medio delle spalle e il naso e tra il punto medio delle spalle e quello delle anche. Tutto questo viene ripetuto per tre volte, per il frame iniziale, quello finale e quello medio.
Inoltre, le varie medie vengono salvate in formato CSV ad ogni sessione riabilitativa di un determinato paziente, per facilitare la successiva unione con quanto estratto dai panel.

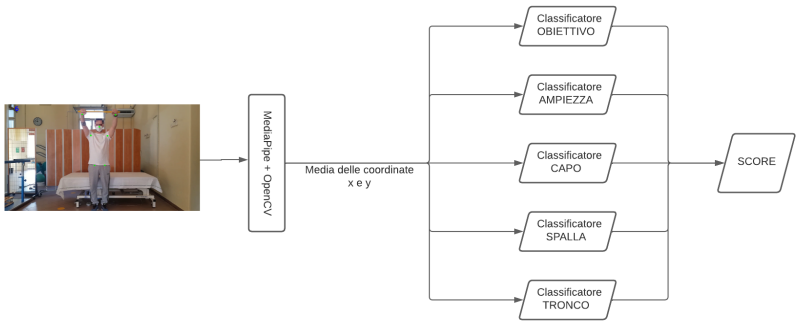

Figura 3.13: Pipeline con tecniche non-deep

Per quanto riguarda, invece, i modelli utilizzati questi sono: Support Vector Machine [37], Random Forest [38], K-Nearest Neighbor [39] e Alberi Decisionali [40].
Per ogni tipologia di esercizio e per ogni aspetto del movimento preso in considerazione sono stati allenati 4 classificatori, uno per ogni tecnica implementata, e successivamente sono state confrontate le loro prestazioni.

Un aspetto molto importante, da cui si può evincere il continuo intervento umano, risulta essere la necessità di inserire manualmente il numero dell'esercizio per il quale si vuole allenare un classificatore oppure semplicemente salvare i punti chiave all'interno di un video.

3.5.1 Modelli Distinti

Il primo framework creato per questo lavoro di tesi è costituito da due elementi fondamentali:

1. estrattore di punti chiave da video tramite MediaPipe e OpenCV e calcolo della media di essi;

2. MLP multi-output per il calcolo della qualità del processo riabilitativo.

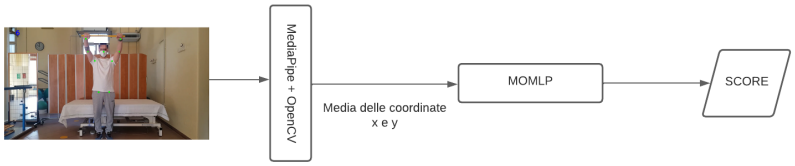

Figura 3.14: Pipeline senza l'utilizzo di Activity Recognition

La manipolazione dei punti chiave utilizza come punto di partenza quanto descritto in 3.5 ma con alcune variazioni.
Le modifiche principali sono essenzialmente due: l'utilizzo per ogni esercizio degli stessi punti chiave (nove) e la mappatura di 48 frame, invece che tre, per eseguire la media degli spostamenti fatti all'interno del video.

I nove punti utilizzati vengono descritti in 3.2.
Partendo dalla totalità dei frame presenti in un video, il cui numero può variare da video a video, e delle rispettive coordinate dei punti chiave, la mappatura avviene creando una struttura dati di 48 elementi. I punti vengono individuati con una semplice proporzione (frame del video * posizione da occupare nella nuova struttura)/48. Una volta ottenuta questa nuova conformazione viene fatta la media utilizzata successivamente come input per il MLP multi-output.
Anche in questo caso i dati estrapolati e manipolati vengono successivamente salvati in file CSV rispetto alle sessioni di riabilitazione eseguite da un paziente.

L'utilizzo di un MLP multi-output elimina la necessità di creare un classificatore per ogni aspetto del movimento da prevedere. Gli input forniti alla rete neurale sono costituiti da 19 features, ovvero le coordinate (x,y) dei 9 punti di riferimento più le coordinate calcolate in riferimento al tronco.
Gli output restituiti dalla rete neurale corrispondo ai cinque aspetti del movimento, descritti in 3.1, i quali successivamente vengono sommati per ottenere la qualità del processo riabilitativo (lo score).
Ovviamente, non avendo nessuna caratteristica che possa identificare la tipologia di esercizio che si sta cercando di valutare resta necessario la creazione di cinque diversi classificatori.

Per ogni esercizio, l'addestramento di un modello mira alla ricerca del numero di layer lineari [41] all'interno di esso, partendo da uno e arrivando a cinque. Ad ogni layer segue un'attivazione, in questo caso si tratta dell'attivazione ReLU [42].
Una caratteristica fondamentale, trattandosi di un MLP multi-output, è la creazione dell'output del modello.
Il layer output è costituito da cinque elementi lineari, ognuno dei quali può assumere un valore tra 0 e 5, a cui viene applicato un Dropout [43] di 0.5.

L'addestramento prevede un learning rate iniziale di 0.1 e un massimo numero di epoche pari a 1000.

Di seguito sono presenti le architetture, con prestazioni migliori, risultate per ogni esercizio.
Per tutti gli esercizi escludendo l'intrarotazione risulta essere un'architettura con cinque layer lineari costituiti da 16, 32, 64, 128, 256 unità.
Esiste una differenza tra l'architetture favorita per elevazione anteriore - elevazione e rotazione superiore (3.15) e abduzione - extrarotazione (3.16) data dalla grandezze dei batch utilizzati per l'addestramento che per i primi due risulta essere di 16 e per i restanti 8.

Per quanto riguarda l'intrarotazione l'architettura "migliore" risulta essere quella costituita da quattro layer lineari composti da 16, 32, 64 e 128 elementi.

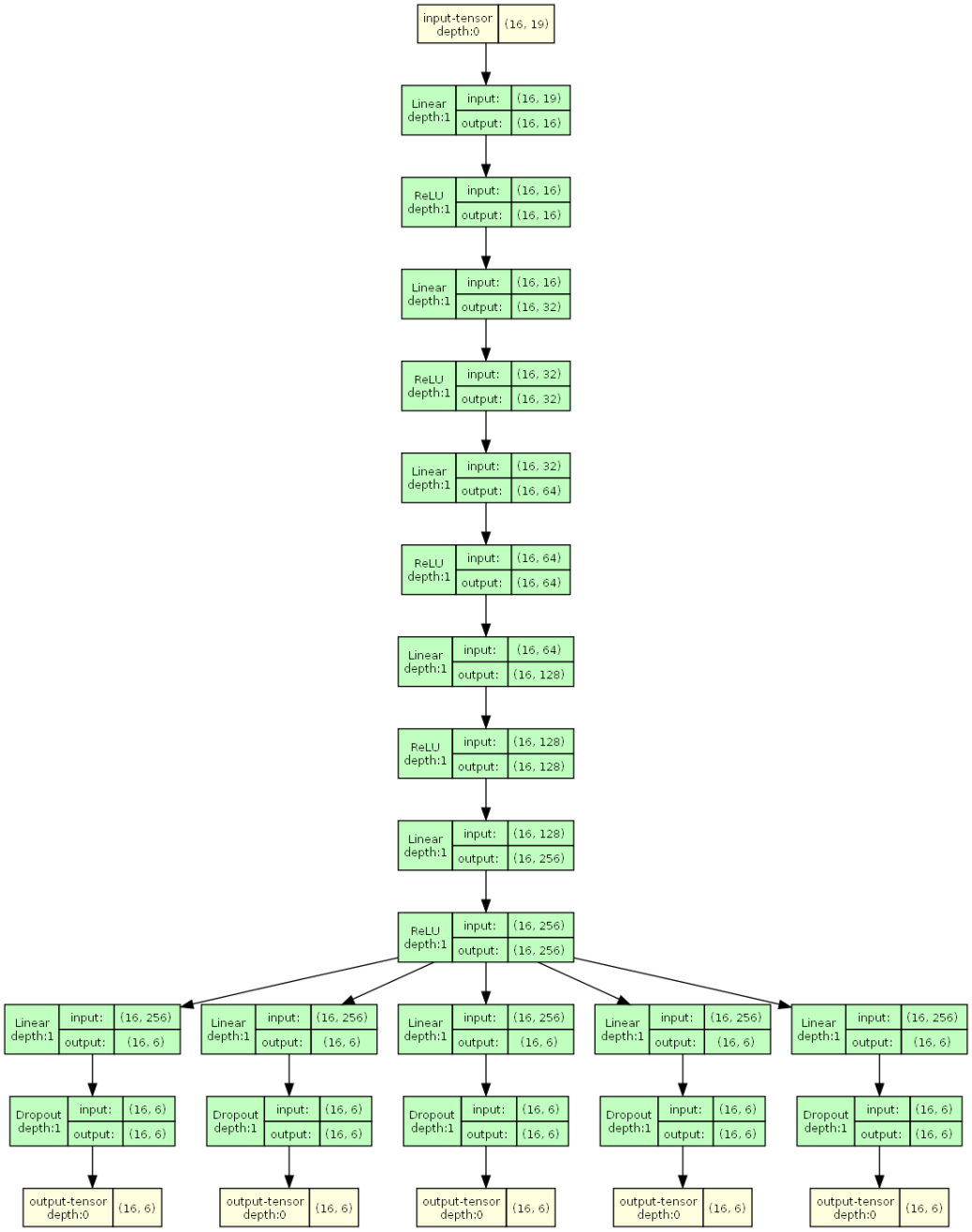

Figura 3.15: Architettura per elevazione anteriore - elevazione e rotazione superiore

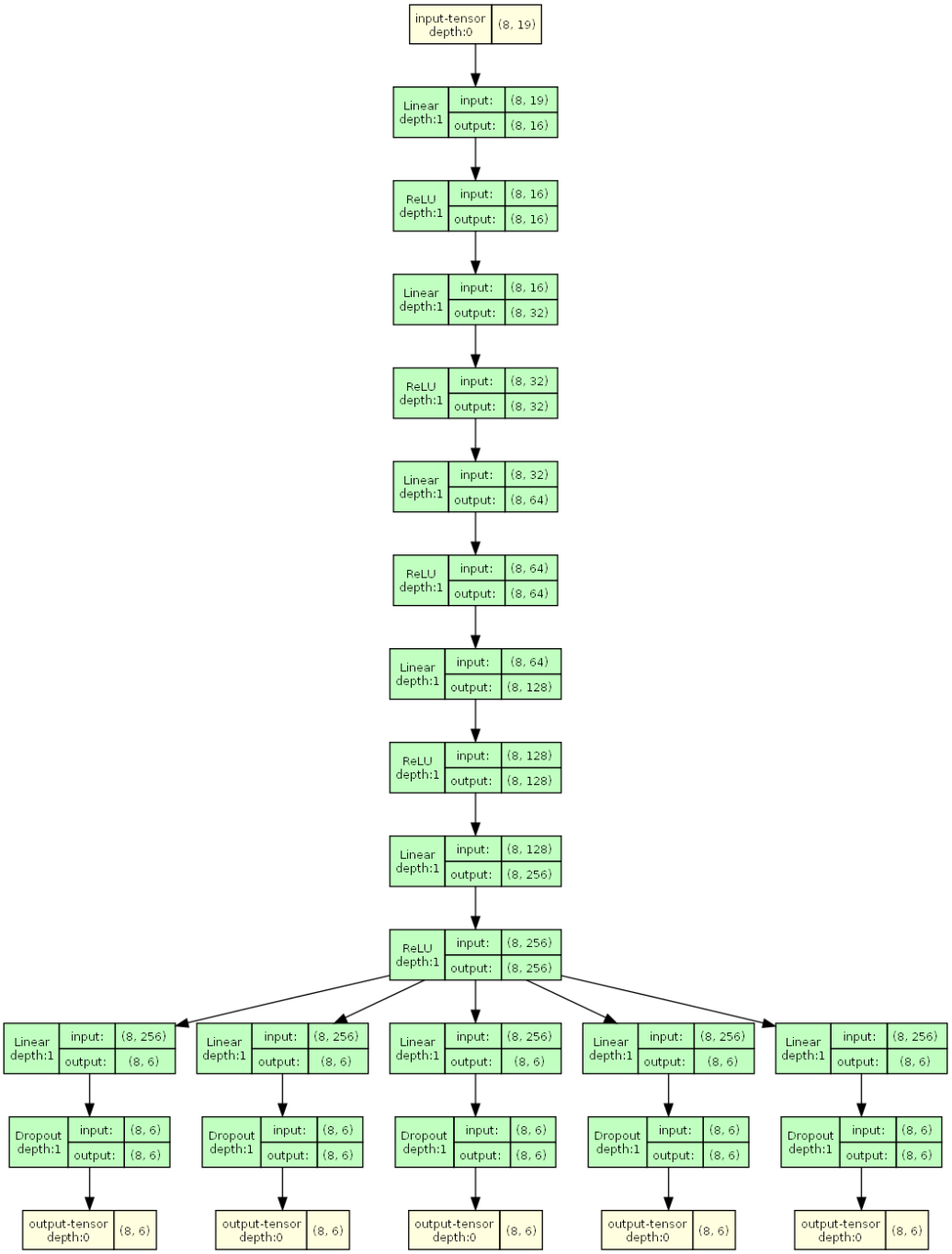

Figura 3.16: Architettura per abduzione - extrarotazione

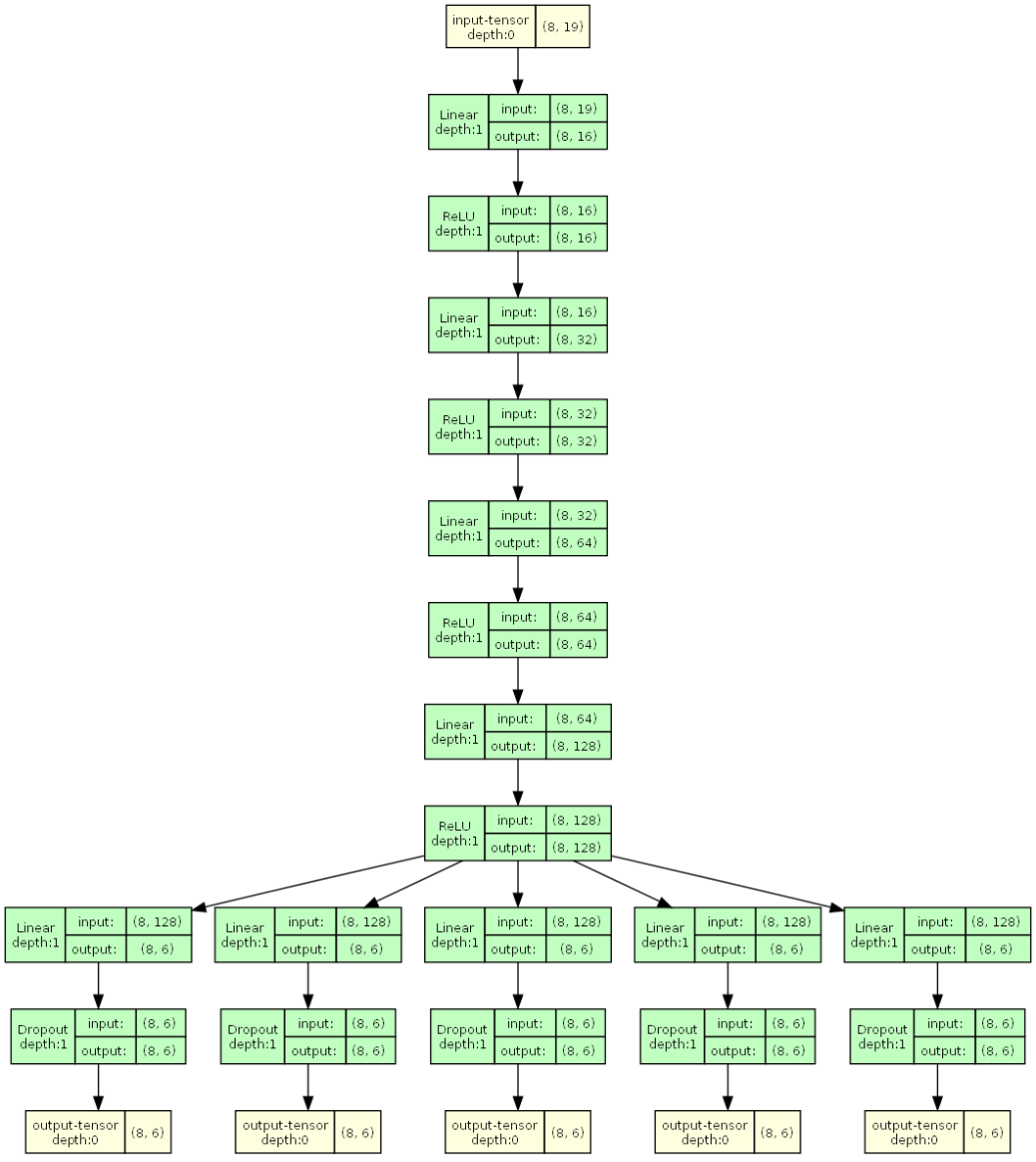

Figura 3.17: Architettura per intrarotazione

3.5.2 Utilizzo di Activity Recognition

Il secondo framework implementato mantiene alcuni aspetti presenti in 3.5.1 ma introduce il riconoscimento dell'esercizio svolto all'interno di un video, perciò il livello di automazione risulta essere più alto.
In particolare, gli elementi che caratterizzano questo secondo approccio sono:

1. estrattore di punti chiave salvando le coordinate corrispettive per 48 frame e media dei punti salvati;

2. utilizzo di un'architettura LSTM per il riconoscimento dell'attività da coordinate di punti chiave di uno scheletro un unico MLP multi-output per il calcolo della qualità del processo di riabilitazione.

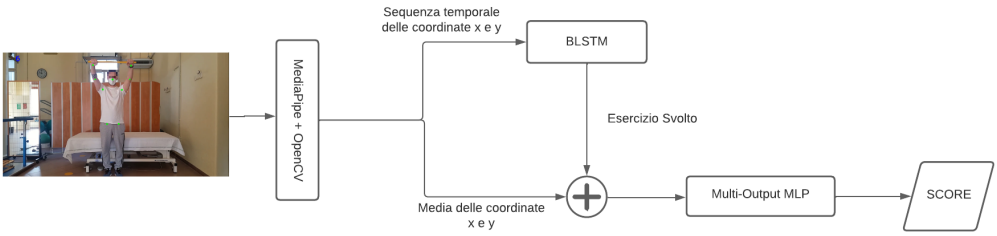

Figura 3.18: Pipeline con Activity Recognition

Il procedimento per la manipolazione dei punti estratti tramite MediaPipe e OpenCV è il medesimo rispetto a quello descritto in 3.5.1.
La differenza fondamentale risulta essere il salvataggio in file CSV non solo della media delle coordinate dei 48 frame di un video ma anche dei singoli 48 frame.
Ciò che accade è una 'conversione' di un video dal formato MP4 ad una sequenza temporale di coordinate (x,y), salvati poi in file CSV, corrispondenti alla posizione dei diversi punti durante l'esecuzione di un esercizio e quindi la riproduzione di un video.
Questo nuovo 'formato' viene utilizzato come input della rete neurale LSTM utilizzata per il riconoscimento dell'attività.

Le classi disponibili sono 5 descritte in 3.1. La dimensione dell'input fornito è data dalla grandezza del batch utilizzato per l'addestramento x 48 x 19.

La ricerca della miglior configurazione per questo modello è stata fatta tramite una grid search degli iperparametri disponibili:

- il numero di unità LSTM (da 1 a 3);
- la grandezza del hidden layer delle unità LSTM(16, 32, 64);
- utilizzo o meno della bidirezionalità di LSTM;
- diverse dimensioni del batch utilizzato per addestrare (8, 16,32).

La configurazione risultante risulta essere composta da una sola unità LSTM ma bidirezionale con un hidden layer contenente 32 elementi.
L'unità LSTM è seguita da un layer lineare e un'attivazione ReLu.

Eseguendo activity recognition non è più necessario creare e allenare modelli differenti rispetto all'esercizio da valutare, questo perchè la tipologia di esercizio eseguita viene fornita, in aggiunta alle features costituite dalla media calcolata durante l'estrazione dei punti chiave, come input al modello che si occupa della classificazione della qualità di un movimento.

Grazie a quanto appena detto il secondo framework prevede un unico MLP multi-output con input di dimensione batch x 20 e 5 output i quali potranno assumere sei classi differenti.

Avendo un output differente rispetto a quanto utilizzato in 3.5.1, viene ripetuta la ricerca della quantità di layer lineari da utilizzare.
In questo caso l'architettura risultante è quella costituita da 5 layer lineari (16,32,64,128,256) ad ognuno dei quali viene applicata un'attivazione ReLU e avente cinque elementi lineari, con dropout di 0.5, che compongono il layer di output .

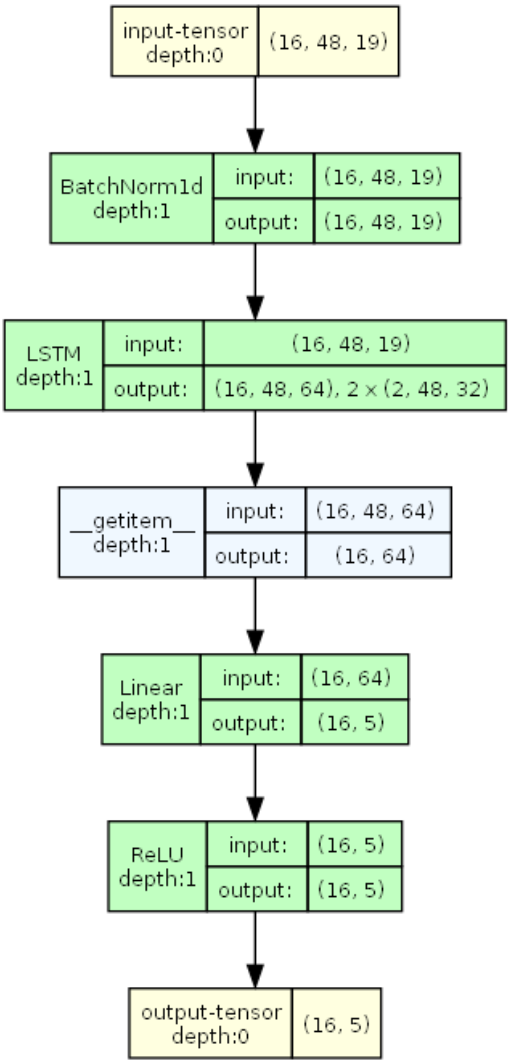

Figura 3.19: Architettura LSTM

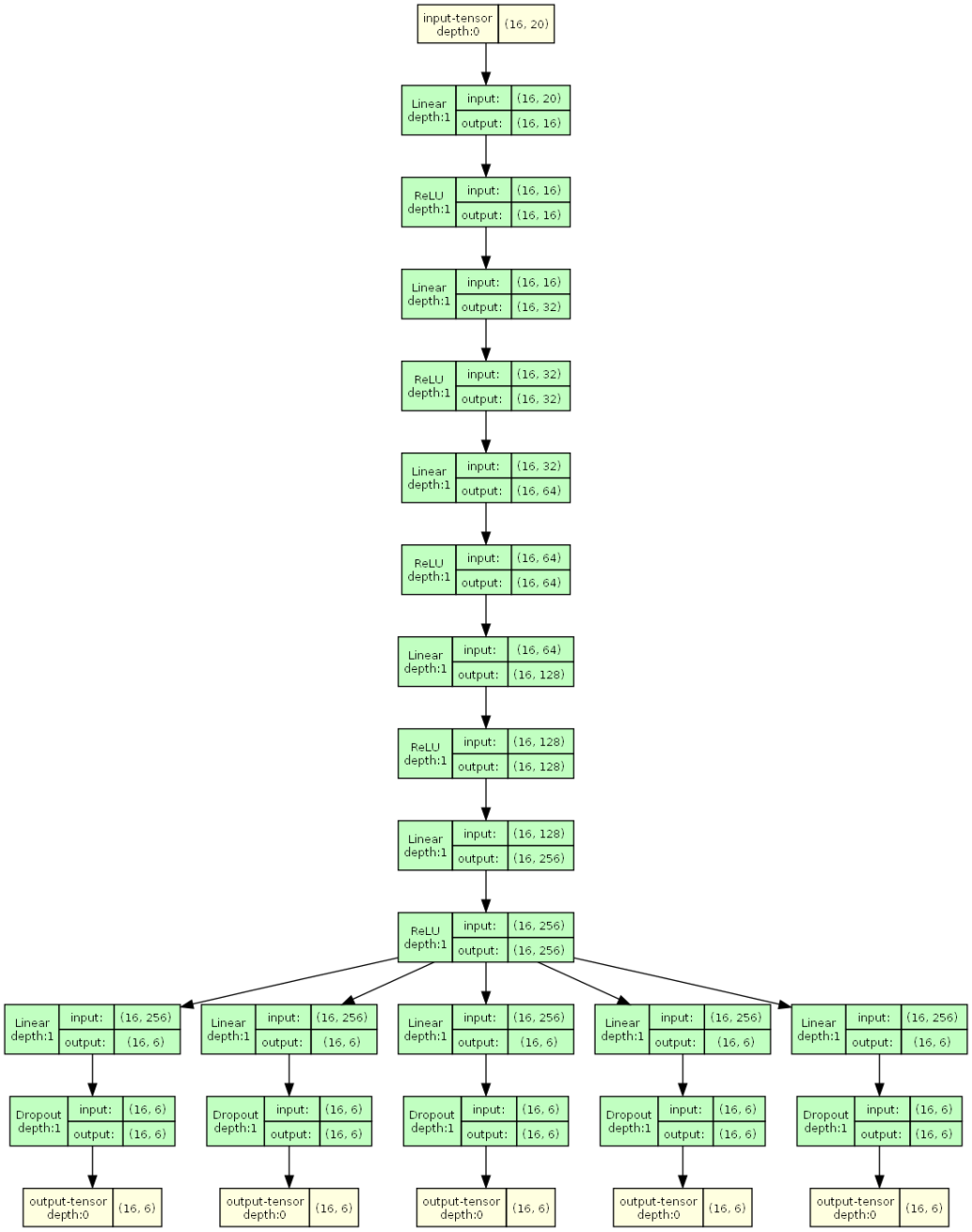

Figura 3.20: Architettura MLP Multi-Output

3.5.3 Architettura End-to-End

L'ultimo step, per ottenere un grado quasi totale di automazione, consiste nella creazione di un'architettura End-To-End.
Con dicitura *End-To-End* si vuole indicare un sistema complesso, composto da moduli differenziabili, il quale viene addestrato tramite l'apprendimento basato sulla discesa del gradiente come un'unica entità. [44]
In questo caso specifico, viene creata un'unica rete neurale composta da due elementi:

1. LSTM per il riconoscimento dell'attività;
2. MLP multi-output o MLP per regressione.

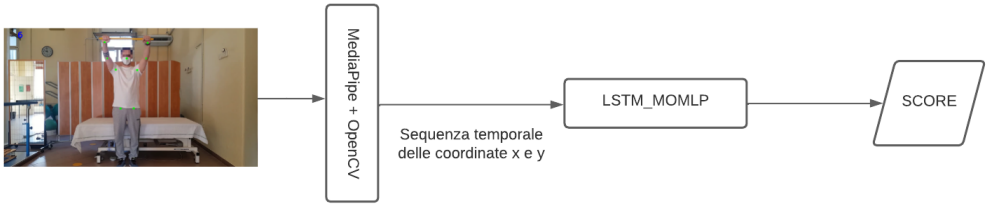

Figura 3.21: Pipeline Classificazione End-to-End

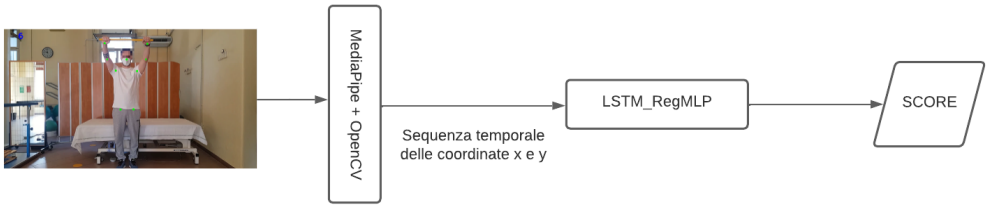

Figura 3.22: Pipeline Regressione End-to-End

Avendo un'unica architettura dove il primo elemento è predisposto per l'activity recognition, durante l'estrazione dei punti chiave del movimento, viene salvata solamente la sequenza temporale di 48 frame.
Perciò l'input dell'intero modello avrà dimensione batch x 48 x 19.

L'output del LSTM, quindi l'esercizio svolto in un video, viene processato durante la fase di addestramento per essere fornite insieme alle features, anch'esse manipolate durante l'addestramento, al MLP (multi-output o regressor) al fine di ottenere la qualità del movimento.

Sono previste due modalità differenti per ottenere l'elaborazione di questi due elementi, durante la fase di forward, perciò due versioni differenti di questa architettura End-To-End.

Versione con Media

Ottenuto l'output del LSTM, la cui dimensione risulta essere un iperparametro, viene calcolata la media della sequenza temporale fornita come input.
La decisione di utilizzare la media deriva da quanto fatto per i framework precedenti dove per l'input fornito al MLP risulta essere una media.

La media e l'attività riconosciuta dal LSTM vengono concatenati, ottenendo un input di dimensione batch x (19 + dimensione dell'output LSTM).
Successivamente è possibile trovarsi davanti a due situazioni differenti, rispetto al modo in cui si vuole ottenere la qualità del processo riabilitativo.
Decidendo di utilizzare un approccio basato sulla classificazione, si avrà un MLP multi-output con le stesse caratteristiche descritte in 3.5.2.
Viene utilizzato però un learning rate iniziale di 0.2

In alternativa a questo tipo di MLP viene implementato un MLP con la finalità di eseguire una regressione che restituisce lo score finale e non più ogni aspetto del movimento precedentemente considerato.

Essendo un nuovo approccio è stata ricercata la miglior configurazione rispetto al numero di layer lineari da utilizzare. Il dropout diventa un iper parametro e nel caso in cui sia utilizzato viene spostato dal layer di output a ogni layer hidden con una probabilità di 0.2.

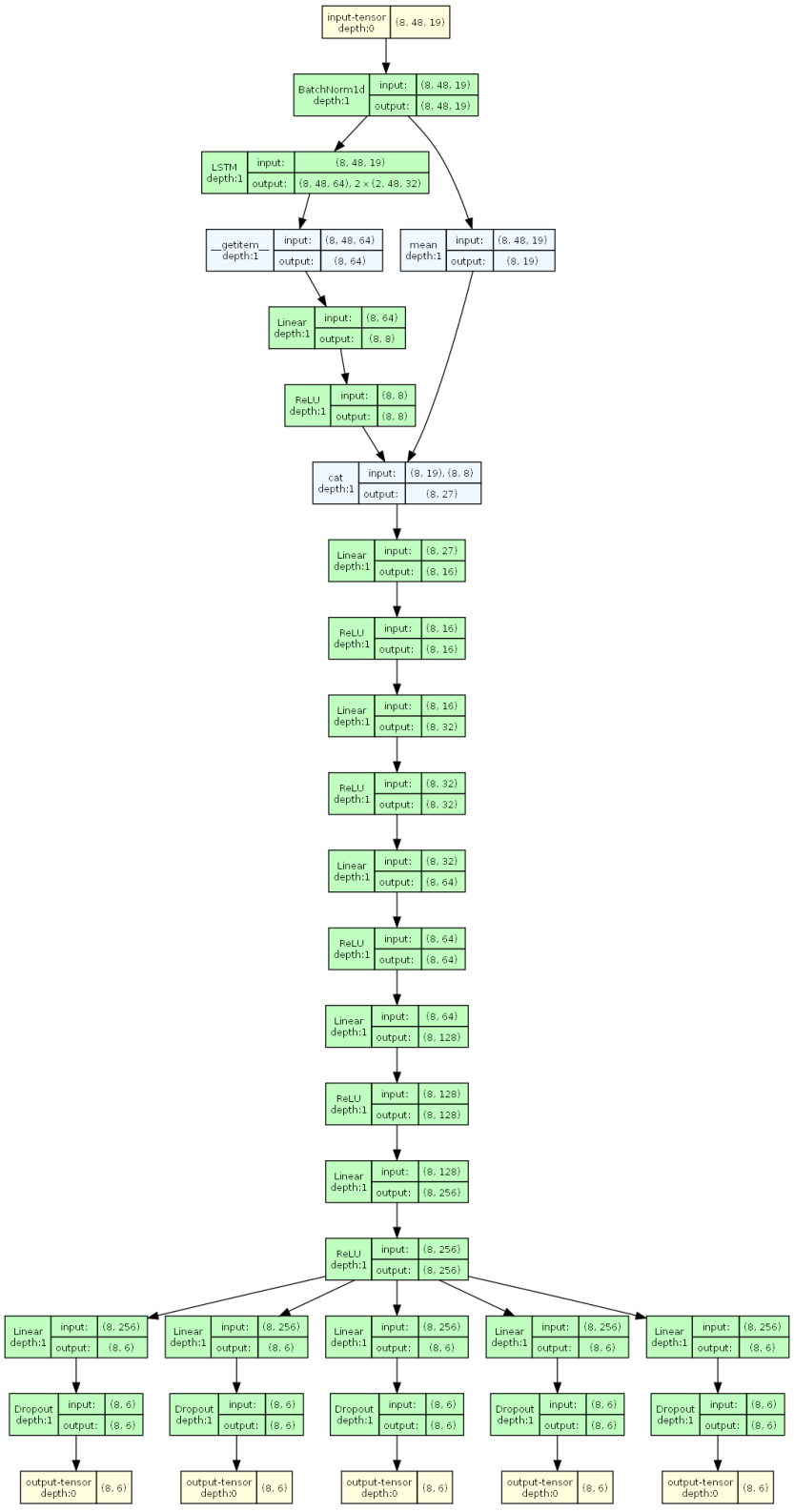

Figura 3.23: Architettura End-to-End Classificazione (v1)

Il learning rate viene abbassato a 0.001 a causa di problemi con il calcolo del gradiente.
L'architettura risultante prevede cinque layer hidden senza utilizzare dropout con una dimensione dell'output di LSTM pari a 5.

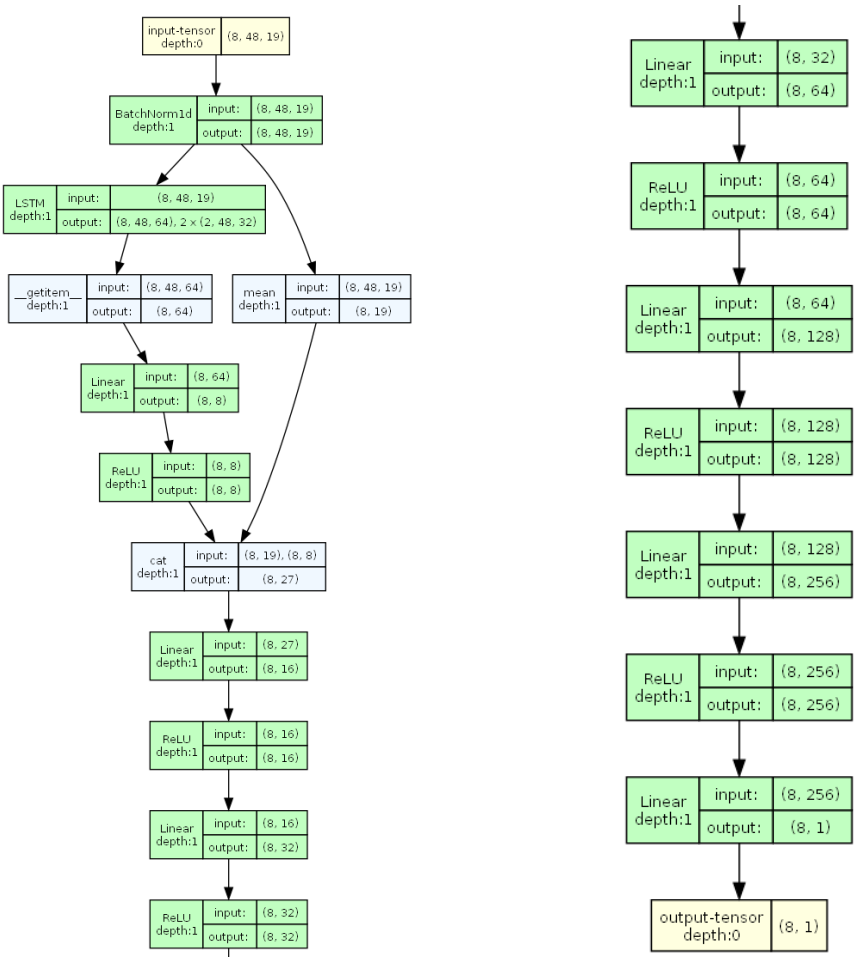

Figura 3.24: Architettura End-to-End Regressione (v1)

Versione con Somma

La seconda modalità di elaborazione dell'output di LSTM e delle feature per la previsione della qualità del movimento sostituisce la media e la concatenazione, precedentemente descritte, con una somma delle due informazioni. Ciò che accade nella fase di forward prevede:

1. il riconoscimento dell'attività svolta, in questo caso avrà una dimensione fissata ovvero 19;
2. l'utilizzo di un layer lineare ricevente l'output di LSTM;
3. la somma delle features in input, senza alcuna manipolazione, con l'output del LSTM;
4. applicazione di un'attivazione ReLu a quanto calcolato;
5. Utilizzo di un ulteriore layer lineare con la corrispettiva.

Successivamente, sia che venga utilizzato un MLP multi-output sia MLP per regressione, vengono utilizzate le configurazioni precedentemente trovare:

- 5 layer lineari con dropout di 0.5 su ognuno di essi,unica differenza con le architetture precedenti, e un layer output con 5 elementi (classificazione);
- 5 layer lineari con dropout di 0.2 su ognuno di essi e un layer output con un elemento(regressione).

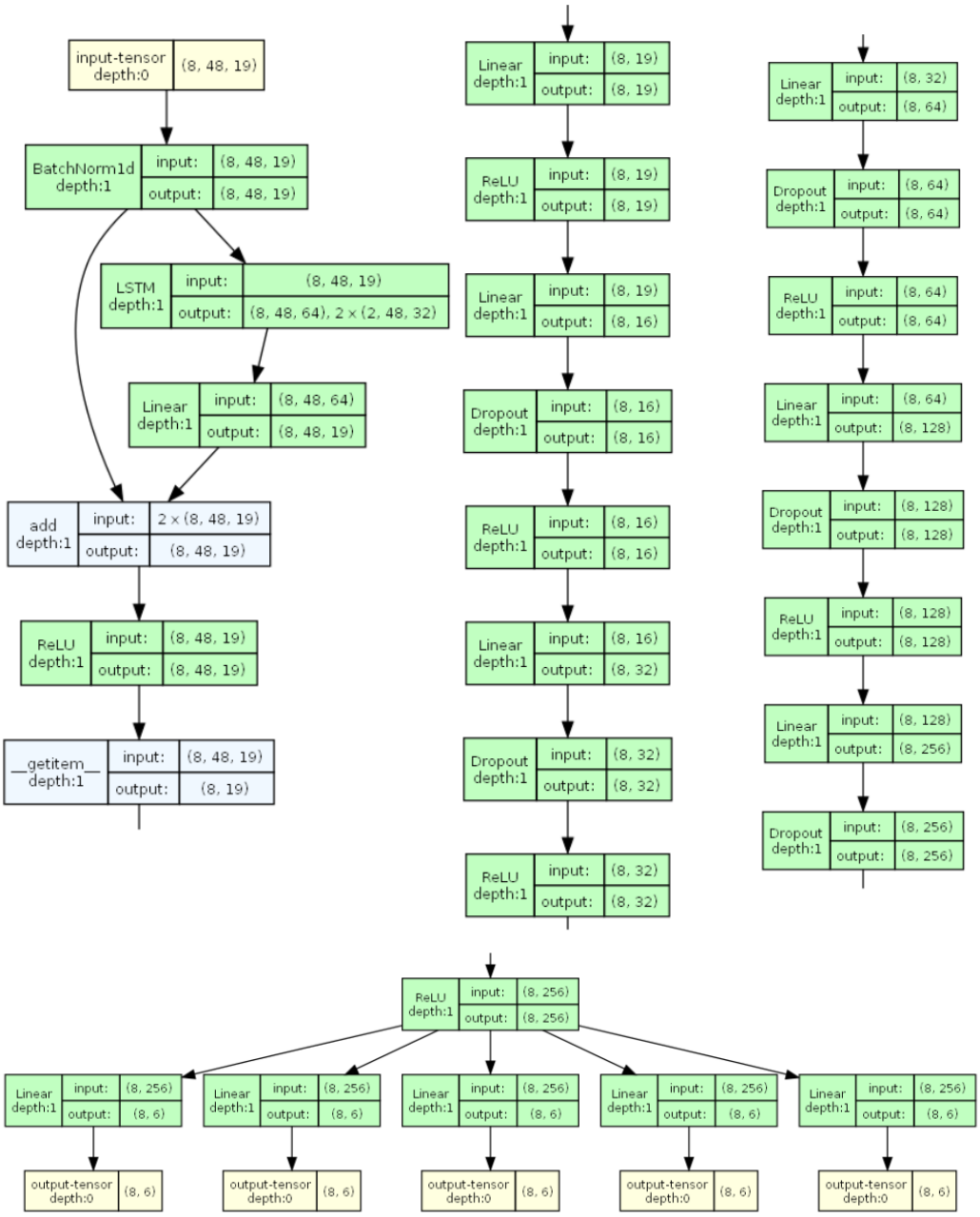

Figura 3.25: Architettura End-to-End Classificazione (v2)

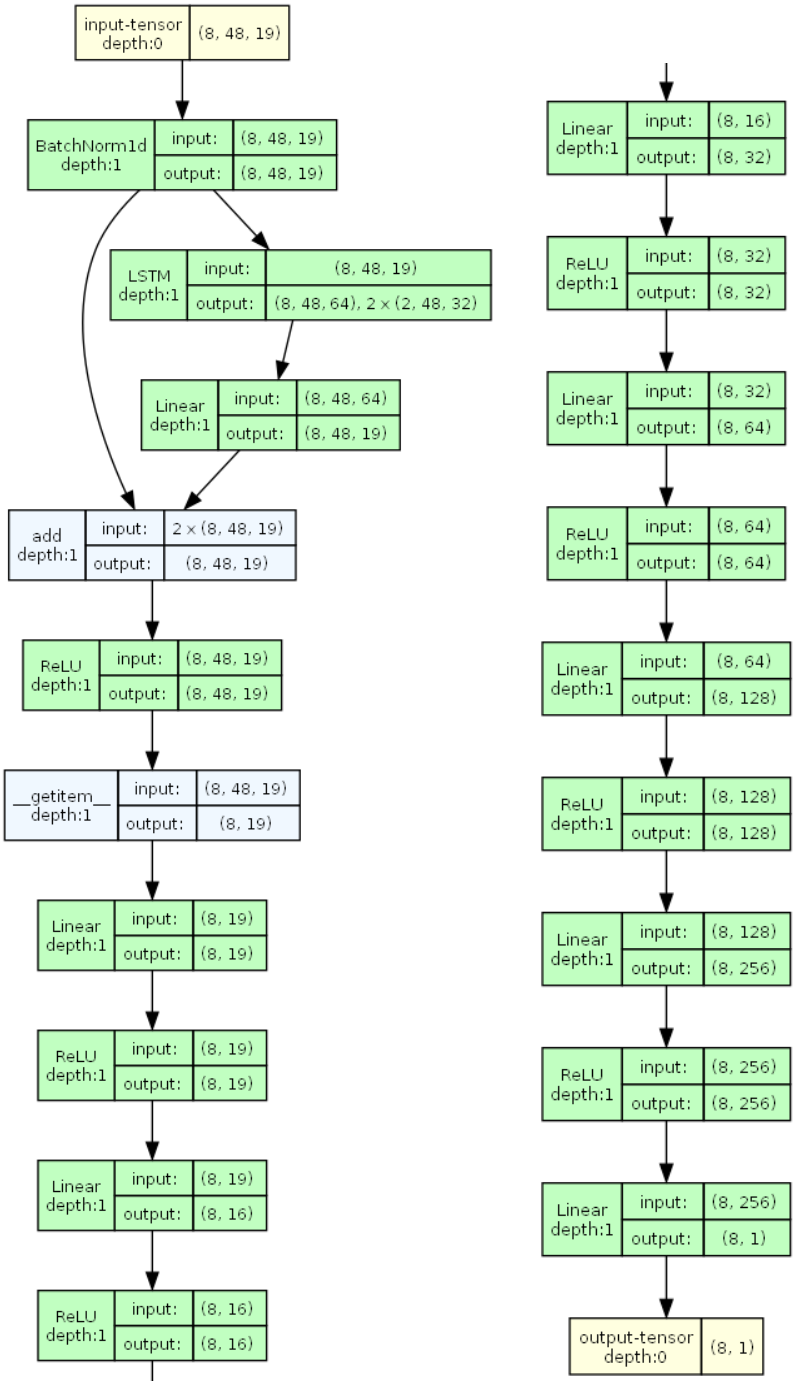

Figura 3.26: Architettura End-to-End Regressione (v2)

Capitolo 4

Risultati Sperimentali e Discussione

All'interno di questo capitolo verranno presentati e confrontati i risultati raggiunti, utilizzando il *best model* ottenuto ad ogni step con la finalità di ottenere una predizione sul set di test estratto dal dataset iniziale.
Verrà posta attenzione anche sugli indici di performance utilizzati per la scelta di questi best model e per poter attuare una comparazione tra i modelli con prestazioni migliori con differente livello di automazione.

Prima di procedere, però, è necessario definire che cosa si vuole indicare con la dicitura *best model*.
Come precedentemente esplicato, all'interno del capitolo 3.5, la fase di training è caratterizzata da Early Stop perciò viene preso in considerazione l'errore ottenuto utilizzando il set di dati attribuito alla validazione.
Ovviamente oltre all'errore vengono calcolate determinate misurazioni, che verranno descritte successivamente.

Nei casi in cui non venga utilizzato un MLP per regressione, le misurazioni e l'errore sono considerati in modo globale, ovvero sommando le misurazioni e gli errori per i singoli aspetti del movimento predetti da un MLP multi-

output e calcolandone la media.

Le misurazioni globali ottenute vengono utilizzate per confrontare le performance tra le varie configurazioni di uno stesso modello al fine di ottenere il cosi detto *best model*: modello con la perdita minore e con indici di performance maggiori, ovviamente sul set di validazione.
Il modello ottenuto viene poi applicato sul set di dati di test ottenendo gli indici di performance utilizzati per il confronto con gli altri livelli di automazione.

Successivamente verranno presentati questi indici di performance, i best model ottenuti per ogni livello di automazione e l'utilizzo dei vari modelli su nuovi dati.

4.1 Indici di Performance

Gli indici di performance utilizzati per effettuare il processo precedentemente descritto sono quattro, ma vengono considerati in situazioni differenti, date dai diversi framework utilizzati: accuratezza, F1 score macro, F1 score weighted e R2 score.

L'utilizzo di LSTM e MLP multi-output prevedono un approccio basato sulla classificazione e quindi condividono l'utilizzo della metrica *accuratezza*, (accuracy) per identificare le performance.
L'accuratezza [45] è un indice che fornisce informazioni sulla percentuale di istanze che il modello ha predetto in modo corretto rispetto alla totalitarietà delle predizioni fatte.

$$Accuratezza = \frac{Numero\ delle\ predizioni\ corrette}{Totale\ delle\ predizioni} \tag{4.1}$$

Nel caso in cui, però, esita uno sbilanciamento tra le possibili classi da prevedere all'interno del dataset è necessario prevedere l'utilizzo anche di altre metriche.

L"accuratezza è stata affiancata dal F1 score, utilizzando due diverse modalità per il suo calcolo.
In generale F1 score [46] è la media armonica tra precision e recall e fornisce una visione complessiva delle performance di un determinato algoritmo di classificazione.

$$F1 = \frac{2 * (precision * recall)}{precision + recall} \tag{4.2}$$

L'apporto fornito da precision e recall ha lo stesso peso nel calcolo di questa metrica.
F1 score può assumere valori all'interno dell'intervallo [0,1]. Per i problemi di classificazione multi-classe, questo indice è dato dalla media del F1 score di ogni classe disponibile sfruttando diversi livelli di ponderazione.
In dettaglio sono stati utilizzati:

- F1 score macro, nel caso dell'utilizzo di bi-LSTM, che calcola la media non pesata per ogni classe non prendendo in considerazione un possibile sbilanciamento. Questo è stato fatto perché la distribuzione della tipologia degli esercizi presenti nei vari video forniti non risulta essere sbilanciata.

- F1 score weighted, nel caso dei singoli MLP multi-output e per le architetture End-to-End, che calcola la media ponderata tra le varie etichette tenendo conto del possibile sbilanciamento tra le classi disponibili. Questo è stato fatto data la quasi assenza di valutazioni basse con rispettiva prevalenza di valutazioni alte.

Per la visualizzazione delle metriche appena citate è stato calcolato e successivamente salvato il report di classificazione [47] per la qualità di ogni aspetto del movimento predetta.
Il report, oltre agli indici esplicitamente utilizzati durante la fase di training

e validation, fornisce informazioni anche su precision e recall.

Column1	precision	recall	f1-score	support
1	1.0	1.0	1.0	18.0
2	0.9615384615384616	1.0	0.9803921568627451	50.0
3	0.9821428571428571	0.9821428571428571	0.9821428571428571	56.0
4	0.98	0.9423076923076923	0.9607843137254902	52.0
5	0.9565217391304348	0.9565217391304348	0.9565217391304348	23.0
accuracy	0.9748743718592965	0.9748743718592965	0.9748743718592965	0.9748743718592965
macro avg	0.9760406115623507	0.9761944577161967	0.9759682133723055	199.0
weighted avg	0.9750599149594125	0.9748743718592965	0.9747758399842349	199.0

Figura 4.1: Esempio Classification Report

Ovviamente avendo previsto la possibilità di ottenere la qualità del processo riabilitativo utilizzando un approccio basato sulla regressione è necessaria l'introduzione di un indice specifico.
Questo indice è R2 score [48], definito anche coefficiente di determinazione, che permette di identificare la forza della relazione tra le variabili indipendenti (features dei movimenti) e la variabile dipendente (score assegnato al movimento) inserite in questo caso in un MLP per effettuare regressione.
La forza della relazione è calcolata utilizzando la varianza considerando diversi aspetti:

- Total sum of squared (TSS), varianza della risposta data dal modello;
- Explained sum of squared (ESS), varianza spiegata dal modello;
- Residual sum of squared(RSS), varianza ancora presente nel modello.

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS} \tag{4.3}$$

L'intervallo all'interno del quale questa metrica può rientrare è [0,1] dove:

- 0 indica l'incapacità delle variabili indipendenti di spiegare la variabilità di y attorno alla sua media;
- 1 indica che le variabili indipendenti sono in grado di spiegare perfettamente la variabilità di y attorno alla sua media.

4.2 Risultati

All'interno del capitolo 3.5 sono state presentate le strutture dei modelli risultanti dalla ricerca del *best model* rispetto ad ogni livello di automazione. Di seguito, invece, verranno mostrati e discussi i risultati e le performance tramite gli indici precedentemente descritti, di queste architetture.

Per quanto riguarda l'utilizzo di modelli separati rispetto all'esercizio da valutare è necessario visionare le singole performance delle cinque architetture addestrate.

Esercizio	**Architettura**	**Accuracy**	**F1-Weighted**
Elevazione Anteriore	multiOutputMLP5	0.8204	0.8204
Abduzione	multiOutputMLP5	0.9532	0.932
Extrarotazione	multiOutputMLP5	0.9434	0.9434
Intrarotazione	multiOutputMLP4	0.9933	0.9933
Elevazione Rotazione Sup	multiOutputMLP5	0.95	0.95

Tabella 4.1: Performance modelli differenti

È possibile notare come il classificatore per la qualità dell'esercizio di elevazione anteriore sia quello con accuratezza e F1-weighted più bassi, entrambi con un valore di 0.8204.
I restanti quattro esercizi hanno tutti performance che rientrano tra 0.94 e 0.99.

Il secondo step per aumentare l'automazione della previsione della qualità del processo riabilitativo prevede l'introduzione del riconoscimento dell'attività svolta all'interno di un video andando ad eliminare la necessità di implementare un modello specifico per ogni esercizio.

Avendo due architetture separate, una per il riconoscimento dell'attività e una per la predizione della qualità, si hanno performance separate.

Modello	Accuracy	F1-Macro
BLSTM	0.9685	0.9676

Tabella 4.2: Performance BiLSTM

Modello	Accuracy	F1-Weighted
Multi-Output MLP	0.9274	0.9274

Tabella 4.3: Performance Multi-Output MLP

Il riconoscimento dell'attività avviene con un'accuratezza di 0.9685 e un F1-macro di 0.9676.

Action	precision	recall	f1-score	support
0	0.9215686274509803	0.9215686274509803	0.9215686274509803	51.0
1	0.9166666666666666	0.9166666666666666	0.9166666666666666	48.0
2	1.0	1.0	1.0	53.0
3	1.0	1.0	1.0	52.0
4	1.0	1.0	1.0	50.0
accuracy	0.968503937007874	0.968503937007874	0.968503937007874	0.968503937007874
macro avg	0.9676470588235293	0.9676470588235293	0.9676470588235293	254.0
weighted avg	0.968503937007874	0.968503937007874	0.968503937007874	254.0

Figura 4.2: Classification Report BiLSTM

Come è possibile notare dal classification report rispetto all'utilizzo del modello sulla parte di dataset utilizzata per la fase di test la 'difficoltà' maggiore è riscontrata nel riconoscimento di elevazione anteriore e abduzione. Questo potrebbe portare a una predizione erronea dell'attività svolta all'interno di un video e quindi ad una successiva previsione non del tutto corretta

della qualità di quel movimento.

Il MLP multi-output risulta raggiungere un'accuratezza globale e un F1-weighted di 0.9274.
Come precedentemente spiegato, si parla di misure di performance globali poiché avendo più output restituiti dallo stesso modello viene fatta la media delle performance rispetto ai vari aspetti del movimento calcolati.
Ovviamente sono stati visionati anche i singoli risultati, tramite classification report, presentanti in seguito.

Column1	precision	recall	f1-score	support
1	0.9642857142857143	0.9	0.9310344827586207	30.0
2	0.9111111111111111	0.8913043478260869	0.9010989010989011	46.0
3	0.855072463768116	0.921875	0.887218045112782	64.0
4	0.9347826086956522	0.9148936170212766	0.9247311827956699	47.0
5	0.95	0.9344262295081968	0.9421487603305784	61.0
accuracy	0.9153225806451613	0.9153225806451613	0.9153225806451613	0.9153225806451613
macro avg	0.9230503795721188	0.9124998388871112	0.9172462744193162	248.0
weighted avg	0.9171334791513612	0.9153225806451613	0.9157136241616501	248.0

(a) Obiettivo

Column1	precision	recall	f1-score	support
1	0.9583333333333334	0.8518518518518519	0.9019607843137256	27.0
2	0.9516129032258065	0.921875	0.9365079365079365	64.0
3	0.8620689655172413	0.9615384615384616	0.9090909090909091	52.0
4	0.9024390243902439	0.925	0.9135802469135802	80.0
5	0.8636363636363636	0.76	0.8085106382978724	25.0
accuracy	0.907258064516129	0.907258064516129	0.907258064516129	0.907258064516129
macro avg	0.9076181180205977	0.8840530626780627	0.8939301030248048	248.0
weighted avg	0.9088380768365998	0.907258064516129	0.9066990407106413	248.0

(b) Ampiezza

Column1	precision	recall	f1-score	support
2	1.0	1.0	1.0	4.0
3	0.9090909090909091	0.7692307692307693	0.8333333333333333	13.0
4	0.9074074074074074	0.8596491228070176	0.8828828828828829	57.0
5	0.9608938547486033	0.9885057471264368	0.9745042492917847	174.0
accuracy	0.9475806451612904	0.9475806451612904	0.9475806451612904	0.9475806451612904
macro avg	0.944348042811773	0.9043464097910558	0.9226801163770002	248.0
weighted avg	0.9465158659946009	0.9475806451612904	0.9464572461065653	248.0

(c) Capo

Column1	precision	recall	f1-score	support
2	1.0	1.0	1.0	6.0
3	1.0	0.8529411764705882	0.9206349206349206	34.0
4	0.8666666666666667	0.9397590361445783	0.9017341040462428	83.0
5	0.959349593495935	0.944	0.9516129032258064	125.0
accuracy	0.9314516129032258	0.9314516129032258	0.9314516129032258	0.9314516129032258
macro avg	0.9565040650406504	0.9341750531537917	0.9439554819767424	248.0
weighted avg	0.9348872279045372	0.9314516129032258	0.9318432695187555	248.0

(d) Spalla

Column1	precision	recall	f1-score	support
2	1.0	0.75	0.8571428571428571	4.0
3	0.9259259259259259	0.9259259259259259	0.9259259259259259	27.0
4	0.9181818181818182	0.9351851851851852	0.9266055045871558	108.0
5	0.9537037037037037	0.944954128440367	0.9493087557603688	109.0
accuracy	0.9354838709677419	0.9354838709677419	0.9354838709677419	0.9354838709677419
macro avg	0.949452861952862	0.8890163098878695	0.9147457608540769	248.0
weighted avg	0.9359570164005648	0.9354838709677419	0.9353895979913889	248.0

(e) Tronco

Figura 4.3: Classification Report MLP Multi-Output con Activity Recognition

Dai report di classificazione è possibile notare come per il capo, spalla e tronco non siano presenti valutazioni inferiori a 2, data la mancanza di dati con questi valori all'interno del dataset utilizzato.
Obiettivo e Ampiezza risultano avere prestazioni inferiori rispettivamente nell'assegnazione della valutazione di 3 e 5.
La predizione di ogni aspetto del movimento però risulta sempre avere un'accuratezza e un F1-weighted compresi tra 0.90 e 0.95, trovando le prestazioni

maggiori nell'assegnazione della valutazione per il capo.

L'ultima categoria di framework implementanti, con architettura End-to-End, risulta avere prestazioni molto differenti rispetto alle due versioni implementate e all'approccio di classificazione e regressione utilizzato.

Partendo dalla regressione, le performance ottenute dalle due architetture presentano alcune differenze ma che conducono ad una stessa conclusione.

Architettura	**Loss**	**R2**
LSTM_RegMLP v1	5.342	0.6023
LSTM_RegMLP v2	0.02053	-0.00024152

Tabella 4.4: Performance Architettura End-to-End Regressione

Come è possibile notare da quanto riportato nella tabella 4.4 la versione che utilizza la media per manipolare l'input fornito al MLP risulta avere una perdita elevata ovvero di 5.342 e un indice R2 di 0.6.
Questo può significare che il modello non sia in grado di adattarsi ai dati, condizione probabilmente dovuta alla mancanza di una vera e propria relazione lineare tra variabili indipendenti e dipendenti.
Questa ipotesi viene avvalorata maggiormente dalla seconda versione implementata che fornisce una loss molto piccola, 0.02053, ma un R2 negativo.
Il fatto che sia presente un R2 di -0.00024152 indica che il modello addestrato non sia totalmente adatto ai dati forniti, adattandosi ad essi in modo errato poiché non segue l'andamento di essi.
È possibile perciò sostenere che utilizzare un approccio basato sulla regressione per ottenere la predizione della qualità del processo riabilitativo, avendo i vincoli derivanti dai dati utilizzati e dall'architettura così implementata, non sia la soluzione più performante indipendentemente dal livello di automazione introdotto. Nel caso della classificazione, le due architetture utilizzate

presentano prestazioni totalmente differenti. È evidente come vi sia una ri-

Architettura	Accuracy	F1-Weighted
LSTM_MOMLP v1	0.9766	0.9766
LSTM_MOMLP v2	0.4905	0.3945

Tabella 4.5: Performance Architettura End-to-End Classificazione

duzione delle performance globali passando dall'utilizzo della media a quello con somma.
La prima versione implementata presenta un'accuratezza e un F1-weighted di 0.9766. La seconda ha performance inferiori, ovvero un'accuratezza di 0.4905 e un F1-weighted di 0.3945.

Per comprendere al meglio come queste misurazioni globali siano state identificate è necessario osservare i classification report dei vari aspetti del movimento predetti da entrambe le architetture multi output.

Column1	precision	recall	f1-score	support
1	1.0	1.0	1.0	34.0
2	1.0	0.9811320754716981	0.9904761904761905	53.0
3	0.9838709677419355	1.0	0.991869918699187	61.0
4	0.9767441860465116	0.9767441860465116	0.9767441860465116	43.0
5	0.9824561403508771	0.9824561403508771	0.9824561403508771	57.0
accuracy	0.9879032258064516	0.9879032258064516	0.9879032258064516	0.9879032258064516
macro avg	0.9886142588278648	0.9880664803738174	0.9883092871145532	248.0
weighted avg	0.987968262226847	0.9879032258064516	0.9879004158705181	248.0

(a) Obiettivo

Column1	precision	recall	f1-score	support
1	1.0	1.0	1.0	28.0
2	1.0	0.9692307692307692	0.9843749999999999	65.0
3	0.9411764705882353	1.0	0.9696969696969697	64.0
4	0.96875	0.96875	0.96875	64.0
5	1.0	0.9259259259259259	0.9615384615384615	27.0
accuracy	0.9758064516129032	0.9758064516129032	0.9758064516129032	0.9758064516129032
macro avg	0.9819852941176471	0.972781339031339	0.9768720862470863	248.0
weighted avg	0.9767552182163188	0.9758064516129032	0.9758327400086473	248.0

(b) Ampiezza

Column1	precision	recall	f1-score	support
2	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	13.0
4	0.9473684210526315	0.9642857142857143	0.9557522123893805	56.0
5	0.9887005649717514	0.9831460674157303	0.9859154929577465	178.0
accuracy	0.9798387096774194	0.9798387096774194	0.9798387096774194	0.9798387096774194
macro avg	0.9840172465060957	0.9868579454253612	0.9854169263367817	248.0
weighted avg	0.9800053715480609	0.9798387096774194	0.9798995227430815	248.0

(c) Capo

Column1	precision	recall	f1-score	support
2	1.0	1.0	1.0	2.0
3	0.96875	0.96875	0.96875	32.0
4	0.9690721649484536	0.9791666666666666	0.9740932642487047	96.0
5	0.9829059829059829	0.9745762711864406	0.978723404255319	118.0
accuracy	0.9758064516129032	0.9758064516129032	0.9758064516129032	0.9758064516129032
macro avg	0.9801820369636092	0.9806232344632768	0.9803916671260059	248.0
weighted avg	0.9758622331369254	0.9758064516129032	0.9758157865725939	248.0

(d) Spalla

Column1	precision	recall	f1-score	support
2	1.0	1.0	1.0	7.0
3	1.0	0.9583333333333334	0.9787234042553191	24.0
4	0.9459459459459459	0.9722222222222222	0.9589041095890412	108.0
5	0.9719626168224299	0.9541284403669725	0.962962962962963	109.0
accuracy	0.9637096774193549	0.9637096774193549	0.9637096774193549	0.9637096774193549
macro avg	0.979477140692094	0.9711709989806321	0.9751476192018307	248.0
weighted avg	0.9641374491766412	0.9637096774193549	0.9637660020189803	248.0

(e) Tronco

Figura 4.4: Classification Report End-to-End V1

Nel caso della prima versione di architettura end-to-end è possibile notare come le performance di predizione siano sempre elevate, indipendentemente da quale aspetto del movimento si prenda in considerazione.

Ovviamente un aspetto importante da considerare è il numero di campioni disponibili che hanno ricevuto una certa valutazione in un certo aspetto. Infatti è possibile notare come per obiettivo e ampiezza il valore 1 venga predetto con una precision, una recall e un F1-score pari a 1, ma considerando il numero di sample rispetto alle altre valutazioni questo risulta essere molto inferiore rispetto alle altre.
La medesima situazione si presenta nel caso del valore 2 per i tre aspetti del movimento rimanenti.
Tutto ciò è dovuto alla scarsità di valutazioni basse all'interno del set di dati utilizzato.

Column1	precision	recall	f1-score	support
1	1.0	1.0	1.0	34.0
2	1.0	0.9811320754716981	0.9904761904761905	53.0
3	0.9838709677419355	1.0	0.9918699186991871	61.0
4	0.9767441860465116	0.9767441860465116	0.9767441860465116	43.0
5	0.9824561403508771	0.9824561403508771	0.9824561403508771	57.0
accuracy	0.9879032258064516	0.9879032258064516	0.9879032258064516	0.9879032258064516
macro avg	0.9886142588278648	0.9880664803738174	0.9883092871145532	248.0
weighted avg	0.9879682622268 47	0.9879032258064516	0.9879004158705181	248.0

(a) Obiettivo

Column1	precision	recall	f1-score	support
1	1.0	1.0	1.0	28.0
2	1.0	0.9692307692307692	0.9843749999999999	65.0
3	0.9411764705882353	1.0	0.9696969696969697	64.0
4	0.96875	0.96875	0.96875	64.0
5	1.0	0.9259259259259259	0.9615384615384615	27.0
accuracy	0.9758064516129032	0.9758064516129032	0.9758064516129032	0.9758064516129032
macro avg	0.9819852941176471	0.972781339031339	0.9768720862470863	248.0
weighted avg	0.9767552182163188	0.9758064516129032	0.9758327400086473	248.0

(b) Ampiezza

Column1	precision	recall	f1-score	support
2	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	13.0
4	0.9473684210526315	0.9642857142857143	0.9557522123893805	56.0
5	0.9887005649717514	0.9831460674157303	0.9859154929577465	178.0
accuracy	0.9798387096774194	0.9798387096774194	0.9798387096774194	0.9798387096774194
macro avg	0.9840172465060957	0.9868579454253612	0.9854169263367817	248.0
weighted avg	0.9800053715480609	0.9798387096774194	0.9798995227430815	248.0

(c) Capo

Column1	precision	recall	f1-score	support
2	1.0	1.0	1.0	2.0
3	0.96875	0.96875	0.96875	32.0
4	0.9690721649484536	0.9791666666666666	0.9740932642487047	96.0
5	0.9829059829059829	0.9745762711864406	0.978723404255319	118.0
accuracy	0.9758064516129032	0.9758064516129032	0.9758064516129032	0.9758064516129032
macro avg	0.9801820369636092	0.9806232344632768	0.9803916671260059	248.0
weighted avg	0.9758622331369254	0.9758064516129032	0.9758157865725939	248.0

(d) Spalla

Column1	precision	recall	f1-score	support
2	1.0	1.0	1.0	7.0
3	1.0	0.9583333333333334	0.9787234042553191	24.0
4	0.9459459459459459	0.9722222222222222	0.9589041095890412	108.0
5	0.9719626168224299	0.9541284403669725	0.962962962962963	109.0
accuracy	0.9637096774193549	0.9637096774193549	0.9637096774193549	0.9637096774193549
macro avg	0.979477140692094	0.9711709989806321	0.9751476192018307	248.0
weighted avg	0.9641374491766412	0.9637096774193549	0.9637660020189803	248.0

(e) Tronco

Figura 4.5: Classification Report End-to-End V2

L'utilizzo della seconda versione dell'architettura End-to-End per eseguire classificazione, così come accade quando si sfrutta la regressione, comporta una riduzione significativa ed evidente delle prestazioni.

Dalle immagini è possibile notare come, per la maggior parte degli aspetti del movimento, il classificatore non sia in grado di prevedere e attribuire correttamente valutazioni di diverso tipo, non solo punteggi bassi.
Questa situazione si verifica, a differenza della prima versione dell'architettura, anche in casi in cui ci sia un numero elevato di sample.

Indipendentemente dall'utilizzo di un approccio basato su classificazione o regressione è evidente che l'utilizzo del metodo che sfrutta la somma per manipolare le informazioni fornite al MLP comporti prestazioni nettamente peggiori rispetto all'utilizzo della media.

Dopo aver preso visione dei risultati e delle prestazioni dei vari framework implementati in questo lavoro di laurea è possibile fare alcune considerazioni.

Innanzitutto utilizzare un approccio basato sulla regressione avendo un'architettura end-to-end caratterizzata, come descritto in 4.2, risulta essere un approccio poco consono rispetto alla predizione della qualità del processo riabilitativo, indipendentemente dal livello di automazione che viene utilizzato.

Considerando i restanti framework, come già specificato, sfruttano uno o più MLP multi-output per fornire predizioni sulla qualità considerando diversi aspetti del movimento.
L'utilizzo di un modello differente per fornire una valutazione per ogni ambito del movimento risulta ottenere prestazioni buone, sui singoli aspetti, ma non introduce alcun tipo di automazione.
L'introduzione di un riconoscimento dell'attività, quindi incrementando l'automazione del processo predittivo, comporta un incremento delle prestazioni globali, visibili nelle metriche globali precedentemente riportare, ma anche nelle prestazioni dell'atTribuzione della valutazione dei singoli aspetti.
Il livello massimo di automazione è raggiunto con l'architettura end-to-end ma le migliori prestazioni sono ottenute dalla prima versione di questa archi-

tettura

Da quanto riportato è possibile, ovviamente rispetto a questo caso di studio, sostenere che l'aumento delle prestazioni e performance vanno di pari passo con l'automazione introdotta da una specifica architettura o modello.

4.3 Demo

Al fine di concludere questo lavoro di tesi e testare il reale funzionamento dei vari framework creati sono state implementate tre differenti demo, una per ogni livello di automazione.

Durante questa fase conclusiva sono stati utilizzati video mai utilizzati prima, senza alcun tipo di valutazione assegnata.
Sono presenti diversi video per ogni tipologia di esercizio, in dettaglio 4 per elevazione e rotazione anteriore e 6 per ognuna delle restanti categorie.
La metà dei video utilizzati è caratterizzata da l'esecuzione totalmente errata degli esercizi al fine di testare se i vari modelli, nonostante la presenza quasi inesistente di valutazioni basse all'interno del dataset utilizzato per il loro addestramento, siano in grado di predire correttamente una qualità del processo riabilitativo non ottima.

Nonostante la diversa architettura che caratterizza ogni modello la pipeline seguita all'interno di questa dimostrazione è la medesima:

1. estrazione dei punti chiave dai video;
2. inizializzazione dei vari modelli e caricamento dei modelli salvati durante la fase di addestramento;
3. predizione della qualità del movimento;
4. salvataggio della predizione in formato CSV.

Ovviamente la diversa conformazione dell'architettura di ogni modello comporta alcune piccole differenze della pipeline sopra descritta.
Esistono differenze nell'estrazione dei punti chiave, seguendo le modalità descritte in 3.5.

Nel caso di utilizzo del riconoscimento dell'attività i modelli istanziati sono due: Bi-LSTM e MLP multi-output. L'output del primo modello verrà poi utilizzato come parte dell'input per il secondo.

Nel caso in cui il riconoscimento dell'attività non è previsto il modello istanziato cambia rispetto al tipo di esercizio da valutare.

Nell'ultima casistica, quella caratterizzata da un'architettura End-to-End, i modelli istanziati sono due dati dalle due diverse modalità con cui viene elaborato l'input fornito al modello.

Di seguito verranno mostrate alcune delle predizione ottenute data l'esecuzione non ottima di extrarotazione.

Obiettivo	Ampiezza	Capo	Spalla	Tronco	Score
2	2	5	4	4	17

(a) Multi Modello

Action	Obiettivo	Ampiezza	Capo	Spalla	Tronco	Score
Extrarotazione	2	2	5	4	4	17

(b) Activity Recognition

Figura 4.6: Predizioni senza e con l'utilizzo di Activity Recognition

Obiettivo	Ampiezza	Capo	Spalla	Tronco	Score
1	3	5	4	4	17

(a) V1

Obiettivo	Ampiezza	Capo	Spalla	Tronco	Score
3	3	5	4	4	19

(b) V2

Figura 4.7: Predizioni architettura End-to-End Classificazione

Come è possibile notare dalle immagini precedenti, tutti e quattro i classificatori hanno attribuito una valutazione elevata per *Capo, Spalla, Tronco*. In questo caso specifico l'utilizzo di modelli separati e l'introduzione di activity recognition al fine di ottenere un unico modello non presentano differenze nella predizione delle varie valutazioni.
Questo non accade considerando le due versione dell'architettura End-to-End.

Nonostante lo score finale risulti essere lo stesso sia per i primi due modelli che per la versione 1, il modo in cui esso viene creato presenta delle differenze, le valutazioni per *Obbiettivo e Ampiezza* subiscono delle variazioni.
La prima versione dell'architettura End-to-End è l'unica in grado di attribuire una valutazione pari a 1.
È evidente come la seconda versione dell'architettura presenti uno score globale maggiore rispetto alle altre tre casistiche dato da una previsione non corretta di *Obbiettivo e Ampiezza*.

Conclusioni e Direzioni Future

Conclusioni

Questo lavoro di tesi si è focalizzato sulla ricerca della migliore configurazione di modelli di Machine Learning per poter effettuare la previsione della qualità del processo riabilitativo in telemedicina dove un aspetto fondamentale risulta essere la correlazione tra performance e automazione del processo predittivo.

Ciò che si voleva dimostrare è il fatto che maggiore è l'automazione fornita da un framework e maggiori sono le prestazioni di predizione.
Lo sviluppo di diversi framework ha permesso di validare questa ipotesi nell'ambito in cui si è operato ovvero valutazione dell'esecuzione di un esercizio di riabilitazione partendo da video.

Il punto di partenza è individuabile nella creazione di modelli differenti in base all'esercizio svolto, i quali forniscono una valutazione sfruttando un MLP multi-output.
Ogni step successivo ha cercato di incrementare il livello di automazione, inserendo il riconoscimento dell'attività e successivamente utilizzando un'architettura end-to-end, al fine di comprendere se ci fosse un'effettiva dipendenza tra questo e la correttezza di quanto predetto.

Quanto trovato in letteratura ha permesso di identificare la strada da intraprendere soprattutto per quanto riguarda il riconoscimento dell'attività partendo da un video, utilizzando la stima della posa di un individuo come sequenza temporale e LSTM come tecnologia.

Un'influenza importante su l'approccio adottato è stata data dal modo in cui è costituita la valutazione attribuita all'esecuzione di un esercizio, la quale considera cinque aspetti diversi di un movimento. Questa caratteristica ha permesso di utilizzare una tecnologia, MLP multi-output, che solitamente non viene utilizzata per fornire una valutazione della qualità del movimento sia in ambito riabilitativo che non, poiché riconducibile ad un singolo valore.

Si è provato ad affrontare la problematica tramite un approccio basato sulla regressione che restituisse una valutazione finale globale, e non cinque valori differenti, al fine di uniformarsi a quanto presente in letteratura. Questo metodo però si è dimostrato avere prestazioni inferiori rispetto a quanto implementato sfruttando approcci basati sulla classificazione.

Sviluppati i diversi framework per avvalorare l'ipotesi di partenza è stato fatto un confronto delle loro prestazioni, tramite diversi indici di performance, e il livello di automazione che li caratterizza.
In conclusione, da questa comparazione, è risultato che il modello costituito da un'architettura addestrata End-to-End presenta le performance migliori, considerando i vari indici di performance, supportate da un livello di automazione massimo rispetto a quanto implementato.

Direzioni Future

Partendo da questa tesi si possono definire e identificare diverse direzioni future e miglioramenti a quanto è stato implementato.

Innanzitutto, considerando l'addestramento del modello, per poter incrementare ulteriormente le prestazioni dell'architettura risultata come 'migliore' è necessario ottenere un incremento del set di dati utilizzato per il suo addestramento, in particolare dati più omogenei dal punto di vista delle valutazioni, al fine di incrementare ulteriormente le sue prestazioni.
Inoltre, eseguire la validazione del modello tramite cross validation porterebbe una maggior credibilità e certezza rispetto ai risultati ottenuti e all'ipotesi sostenuta.

Un lavoro futuro interessante potrebbe prevedere l'utilizzo, in aggiunta a quanto già utilizzato, di features relative alle caratteristiche di una persona, ad esempio età e condizione fisica, per poter ottenere la predizione della qualità del processo riabilitativo.

Un'altra direzione futura implementabile da quanto già sviluppato è la predizione della qualità di un esercizio in tempo reale con l'aggiunta di feedback per la correzione di quanto si sta eseguendo. Questo feedback può essere sia testuale che visivo, mostrando durante l'esecuzione errata di un determinato esercizio la corretta esecuzione tramite la sovrimpressione di uno scheletro creato tramite la stima della posa.

Infine, quanto sviluppato potrebbe essere inserito all'interno di una Web App creata appositamente per monitorare il processo riabilitativo di un individuo sia da parte di medici che da parte dei pazienti stessi.

Bibliografia

[1] World Health Organization. (2017). *The need to scale up rehabilitation.* World Health Organization.

[2] D.R. Masys, *Telemedicine: A Guide to Assessing Telecommunications in Health Care*, Journal of the American Medical Informatics Association, Vol. 4, pp. 136–137, 1997

[3] A. Dahiya, B. Charlot, M. Dhifallah, T. Gil, N. Azémard, A. Todri-Sanial, J. Thireau, A. Lacampagne, J. Boudaden, P. Ramm, T. Kießling, A. Fraunhofer, S. Lal, C. O'Murchu, K. Razeeb, U. Gulzar, Y. Zhang, EC Workshop on 'Smart Bioelectronic and Wearable Systems' ,*SmartVista: Smart Autonomous Multi Modal Sensors for Vital Signs Monitoring*, 2019

[4] F. Aberer, D.A. Hochfellner, J.K. Mader, *Application of Telemedicine in Diabetes Care: The Time is Now*, Diabetes Ther 12, pp. 629–639, 2021

[5] M. Zampolini, E. Todeschini, G. Montserrat, H Hermens, S. Ilsbroukx, V. Macellari, R. Magni, M. Rogante, S. Marchese, M. Vollenbroek - Hutten, C. Giacomozzi, *Tele-rehabilitation: Present and future*, Annali dell'Istituto superiore di sanità, vol. 44, pp. 125-34, 2008

[6] A. Peretti, F. Amenta, S. Tayebati, G. Nittari, S. Mahdi, *Telerehabilitation: Review of the State-of-the-Art and Areas of Ap-*

plication, JMIR Rehabilitation and Assistive Technologies, vol. 4, pp. e7, 2017

[7] L. Cacciante, C.d. Pietà , S.Rutkowski, et al, *Cognitive telerehabilitation in neurological patients: systematic review and meta-analysis.*, Neurol Sci 43, pp. 847–862, 2022

[8] K. J. Davis, D. Pagliuco, *Chapter 23 - Telerehabilitation in Speech-Language Pathology*, pp. 339-349, 2022

[9] R.W.M. Brouwers, H.J. van Exel, J.M.C. van Hal, et al, *Cardiac telerehabilitation as an alternative to centre-based cardiac rehabilitation*, Neth Heart J 28, pp. 443–451, 2020

[10] Y. Liao, A. Vakanski, M. Xian, D. Paul, R. Baker, textitA review of computational approaches for evaluation of rehabilitation exercises, Computers in Biology and Medicine, vol. 119, 2020

[11] A.E. Tozzi, *Il connubio tra telemedicina e intelligenza artificiale per un salto di qualità nelle cure*, Agenzia Nazionale per i Servizi Sanitari Regionali, 2021

[12] Z. Zhao, S. Kiciroglu, H. Vinzant, Y. Cheng, I. Katircioglu, M. Salzmann, P. Fua, *3D Pose Based Feedback for Physical Exercises*, Proceedings of the Asian Conference on Computer Vision (ACCV), pp. 1316-1332, 2022

[13] Y. Liao, A. Vakanski, M. Xian, *A Deep Learning Framework for Assessing Physical Rehabilitation Exercises*, IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 28, no. 2, pp. 468-477, Feb. 2020

[14] A. Miron, C. Grosan, *Classifying action correctness in physical rehabilitation exercises*, CoRR, Aug. 2021

[15] J. Francisco; P. Rodrigues, *Computer Vision Based on a Modular Neural Network for Automatic Assessment of Physical Therapy Rehabilitation Activities*, IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 31, pp. 2174-2183, 2023

[16] K. Sarker, Mohamed Masoud, Saeid Belkasim, Shihao Ji, *Towards Robust Human Activity Recognition from RGB Video Stream with Limited Labeled Data*,17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 145-151, 2018

[17] F. Noori, B. Wallace, Z. Uddin, J. Torresen, *A Robust Human Activity Recognition Approach Using OpenPose, Motion Features, and Deep Recurrent Neural Network*, 21st Scandinavian Conference - SCIA, pp. 299–310 Jun. 2019

[18] C. Li, P. Wang, S. Wang, Y. Hou, W. Li, *Development of Action-Recognition Technology Using LSTM Based on Skeleton Data*, 9th International Conference on Industrial Application Engineering, 2021

[19] *OpenCV*, https://opencv.org

[20] *MediaPipe*, https://developers.google.com/mediapipe

[21] Zhang, Aston and Lipton, Zachary C. and Li, Mu and Smola, Alexander J. (2023). *Dive into Deep Learning.* Cambridge University Press. https://d2l.ai/index.html

[22] S. Hochreiter and J. Schmidhuber, *Long Short-Term Memory*, Neural Computation, vol. 9, no. 8, pp. 1735-1780, Nov. 1997

[23] A. Graves and J. Schmidhuber , *Framewise phoneme classification with bidirectional LSTM networks*, IEEE International Joint Conference on Neural Networks, pp. 2047-2052 vol. 42005, 2005

[24] F.A. Gers, J. Schmidhuber, F. Cummins, *Learning to forget: continual prediction with LSTM.*, Ninth International Conference on Arti-

ficial Neural Networks ICANN 99. (Conf. Publ. No. 470), pp. 850-855 vol.2, 1999

[25] F. Gers, J. Schmidhuber, *Recurrent nets that time and count*, International Joint Conference on Neural Networks,pp. 189-194 vol.3, 2000

[26] G. Van Hout, C. Mosquera, G. Napoles, *A review on the long short-term memory model*, Artificial Intelligence Review, Volume 53, Issue 8, pp 5929–5955, Dec. 2020

[27] L. Songa, G. Yub, J. Yuana, Z. Liuc, *Human Pose Estimation and Its Application to Action Recognition: A Survey*, Journal of Visual Communication and Image Representation, Volume 76, 2021

[28] Rosenblatt, F. (1958). *The Perceptron: A Theory of Statistical Separability in Cognitive Systems* (Project PARA). Washington: U.S. Dept. of Commerce, Office of Technical Services

[29] *PyTorch Documentation*, https://pytorch.org/docs/stable/index.html

[30] *CrossEntropyLoss*, https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html#torch.nn.CrossEntropyLoss

[31] *MSELoss*, https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html#torch.nn.MSELoss

[32] *Stochastic Gradient Descent*, https://scikit-learn.org/stable/modules/sgd.html

[33] *Exponenstial LR*, https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ExponentialLR.html?highlight=torchoptimlr_scheduler

[34] *Batch Normalization*, https://pytorch.org/docs/stable/generated/torch.nn.BatchNorm1d.html?highlight=batch+normalization

[35] *Scikit-learn*, https://scikit-learn.org/stable/index.html

[36] *train_test_split*, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[37] *Support Vector Machine*, https://scikit-learn.org/stable/modules/svm.html#svm

[38] *Random Forest Classifier*, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier

[39] *KNeighborsClassifier*, https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn-neighbors-kneighborsclassifier

[40] *Decision Tree Classifier*, https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[41] *Linear*, https://pytorch.org/docs/stable/generated/torch.nn.Linear.html

[42] *ReLU*, https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html

[43] *Dropout*, https://pytorch.org/docs/stable/generated/torch.nn.Dropout.html

[44] T. Glasmachers, *Limits of End-to-End Learning*, CoRR, 2017

[45] *accuracy_score*, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

[46] *f1_score*, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

[47] *classification_report*, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

[48] *r2_score*, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html

Ringraziamenti

Grazie a chi c'è sempre stato e a chi se né andato.
Grazie a chi ha creduto in me fin dall'inizio e fino alla fine.
Grazie a chi oggi non c'è più ma è sempre con me.