

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZA E INGEGNERIA · SEDE DI BOLOGNA
DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA
- DISI -

Pose estimation per identificare e valutare esercizi statici a corpo libero

LAUREA TRIENNALE IN
INFORMATICA

RELATORE:

**Char.mo Prof.
Andrea Asperti**

PRESENTATA DA:

Lorenzo Girotti

CO-RELATORE:

**Dr.
Giorgio Tsiotas**

SESSIONE I

Anno Accademico 2023/2024

Abstract

Questa tesi propone un approccio alternativo per affrontare la sfida della verifica dell'accuratezza nell'esecuzione di alcuni esercizi di forza a corpo libero. L'obiettivo principale è quello di individuare l'esercizio eseguito, determinarne la correttezza e fornire una panoramica degli errori commessi durante l'esecuzione di tale elemento, questo mediante algoritmi di *Computer Vision* e *Deep Learning* secondo l'attuale stato dell'arte.

Nel contesto sportivo, diverse applicazioni hanno cercato di affrontare sfide simili. Tuttavia, la soluzione proposta in questa tesi vuole garantire accessibilità e flessibilità, questo approccio infatti si discosta dall'utilizzo di strumenti specifici e costosi, come sensori 3D, o da configurazioni impegnative, come il posizionamento preciso di più fotocamere.

Indice

1	Introduzione	7
2	Background teorico	9
2.1	Computer vision	9
2.1.1	Object detection	10
2.1.2	Segmentazione semantica	11
2.2	Pose estimation	12
2.2.1	Pose estimation 2D	13
2.2.2	Pose estimation 3D	15
2.3	Applicazioni nello sport	18
2.3.1	Salute e sport	19
2.3.2	Strategie e sport di squadra	19
2.3.3	Tecniche di allenamento e sport individuali	20
2.4	Ginnastica artistica e Calisthenics	20
3	Il progetto	23
3.1	Tecnologie utilizzate	23
3.1.1	Python	23
3.1.2	Tensorflow	24
3.1.3	Colab	25
3.1.4	MediaPipe Pose Landmark Detection	25
3.2	Creazione e pre-processamento del dataset	26
3.2.1	Raccolta dei dati	26
3.2.2	Pre-processing	27
3.3	Specifiche dei modelli	27
3.3.1	Modello Dense	27
3.3.2	Modello CNN	28
3.4	Addestramento dei modelli	28
3.4.1	Suddivisione del dataset	28
3.4.2	Fase di training	29
3.5	Considerazioni e confronto dei modelli	30
3.6	Calcolo della correttezza di un esercizio	33

4 Conclusioni	37
4.1 Sviluppi futuri	37
4.1.1 Allargamento del dataset e riconoscimento di nuovi elementi .	37
4.1.2 Classificazione tramite video e riconoscimento di elementi dinamici	38
4.1.3 Adattamento della valutazione a prospettive diverse	38
4.2 Applicazioni reali	38

Capitolo 1

Introduzione

Negli ultimi anni, con l'importante sviluppo del *Deep Learning*, il campo della *Computer Vision* ha vissuto una trasformazione senza precedenti. L'applicazione di queste tecnologie alle immagini ha reso possibile una vasta gamma di applicazioni[1], tra cui il riconoscimento di oggetti, il riconoscimento facciale, il rilevamento di azioni e la stima della posa umana.

Tra i numerosi ambiti di applicazione possiamo trovare l'impiego di questi strumenti nello sport[2]. Infatti grazie alle tecnologie di visione artificiale, è possibile analizzare dettagliatamente le performance degli atleti sia negli sport di squadra che negli sport individuali. Un esempio è la possibilità di valutare le proprie strategie di gioco per individuare eventuali carenze, oppure analizzare gli avversari per comprendere come contrastarne efficacemente le tattiche.

In questa tesi si vuole analizzare singolarmente alcuni degli esercizi statici di forza tipici degli sport a corpo libero come *Ginnastica artistica* o *Calisthenics*¹, alcuni di questi eseguibili su più attrezzi diversi come anelli, parallele o a terra. In seguito, dopo aver riconosciuto l'esercizio, si vuole andare a valutare quali possono essere i possibili errori nell'esecuzione (es. corpo non parallelo al terreno, braccia non tese...) tramite la rilevazione e l'analisi dei punti chiave del corpo dell'atleta in oggetto.

Nel contesto della ginnastica artistica, alcuni studi hanno già implementato tecniche di Deep Learning. Già nel 2018, Fujitsu ha sviluppato un sistema ICT per valutare le performance degli atleti durante le loro routine[3]. Tale tecnologia è stata estesa per generare giudizi su tutte le possibili tipologie di esercizi eseguibili dagli atleti su vari attrezzi ginnici, sia per la categoria maschile che per quella femminile. Tuttavia, a differenza dell'approccio proposto in questa tesi, il sistema citato richiede un computer ad alte prestazioni e l'utilizzo di un sensore laser 3D per applicare tecniche di pose estimation.

¹Disciplina basata sulla ginnastica artistica

Con questa tesi si vuole mostrare un metodo che assicuri la flessibilità di valutare l'esecuzione di un esercizio attraverso un video o un'immagine, sfruttando semplicemente uno smartphone, ampliando notevolmente le opportunità di valutazione e autovalutazione. Questo approccio inoltre mira a facilitare il raggiungimento rapido e sicuro degli obiettivi sportivi, rendendo accessibile anche in autonomia la verifica della correttezza, senza aver necessariamente bisogno della presenza costante di un allenatore.

Capitolo 2

Background teorico

2.1 Computer vision

La computer vision è un campo dell'informatica che ha come obiettivo la comprensione e l'analisi di immagini o video. Utilizzando tecniche di machine learning e deep learning, la computer vision permette ai calcolatori di interpretare e comprendere il significato di un qualunque contenuto multimediale visivo.

Le applicazioni della computer vision sono molte e ricoprono settori come la sicurezza, la medicina, il controllo della qualità, fino all'assistenza alla guida autonoma o al campo dell'edilizia[4]. Un esempio di tecnologia diffusa che incontriamo spesso nella quotidianità è il riconoscimento facciale, funzionalità ormai presente nella quasi totalità degli smartphone.

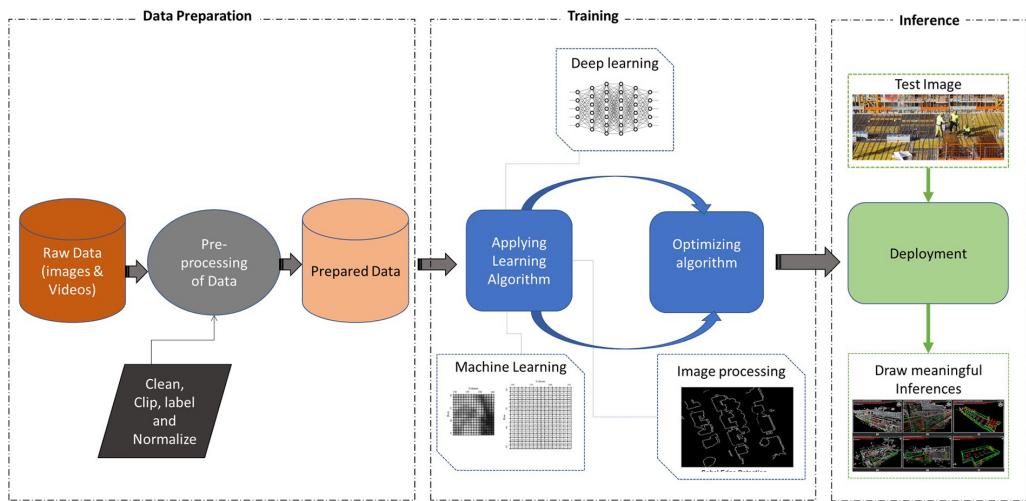


Figura 2.1: Schema del processo di elaborazione tipico della computer vision[4]

Di seguito, un approfondimento su due degli argomenti più studiati all'interno della computer vision: *object detection* e *semantic segmentation*. Questi due argomenti, costituiscono le fondamenta per la comprensione di altre attività parallele,

alcune di queste sono l'estrazione di features, il riconoscimento di pattern, il tracking di oggetti e la stima della posa.

2.1.1 Object detection

L'object detection è un'importante tecnica che rientra nell'ambito della computer vision, si concentra sull'individuare e localizzare gli oggetti all'interno di un contenuto multimediale. Questo tema è stato al centro delle ricerche degli ultimi 20 anni, seguendo una crescita esponenziale, nel 2021 sono state quasi 3500 le pubblicazioni che hanno trattato questo argomento[5].

In generale, l'identificazione di oggetti ha come obiettivo individuare, classificare e mostrare il grado di confidenza con cui ha rilevato uno o più oggetti all'interno di un'immagine. E' bene sottolineare la presenza di due diverse tecniche di identificazione, *region proposals* e *single shot*.

Region proposals Seguendo questo approccio si cerca inizialmente di individuare le potenziali aree dell'immagine o del frame che potrebbero contenere un oggetto. Successivamente, una volta individuate queste regioni, si procede con l'effettiva parte di identificazione. In questa fase, tramite una tecnica bottom-up, vengono tracciati dei rettangoli che delimitano gli oggetti all'interno delle regioni selezionate, riducendo così lo spazio di ricerca[6]. Un modello di rete neurale specialmente diffuso nell'implementazione di tecniche region proposals è R-CNN[7]. Questa tipologia di rete opera suddividendo inizialmente l'immagine in 2000 regioni. In seguito, per ridurre il numero di regioni e considerare solo quelle più significative, viene utilizzato un algoritmo di tipo selective search. Come ultimo passo, per determinare la presenza di un oggetto, le regioni selezionate vengono inserite in una Support Vector Machine (SVM)[8].

Esistono anche versioni ottimizzate di R-CNN: Fast R-CNN[9] e Faster R-CNN[10].

Single shot Per quanto riguarda le tecniche single shot, esistono due tipologie di implementazioni tipicamente utilizzate: You Only Look Once (YOLO)[11] e Single Shot Detector (SSD)[12]. Utilizzando YOLO, vengono inizialmente individuate le zone in cui è più probabile che ci siano oggetti da identificare, questo tramite una singola valutazione dell'intera immagine da analizzare, da cui deriva il nome dell'architettura. Le zone identificate vengono poi delimitate tramite rettangoli, fornendo anche un valore di confidenza associato alla predizione. Questo valore rappresenta la sicurezza con cui YOLO ritiene che un determinato rettangolo contenga effettivamente l'oggetto che ha identificato[13]. Inoltre, è importante sottolineare la velocità di YOLO, sfruttando una singola valutazione riesce infatti a performare fino a 100 volte più velocemente di Fast R-CNN.

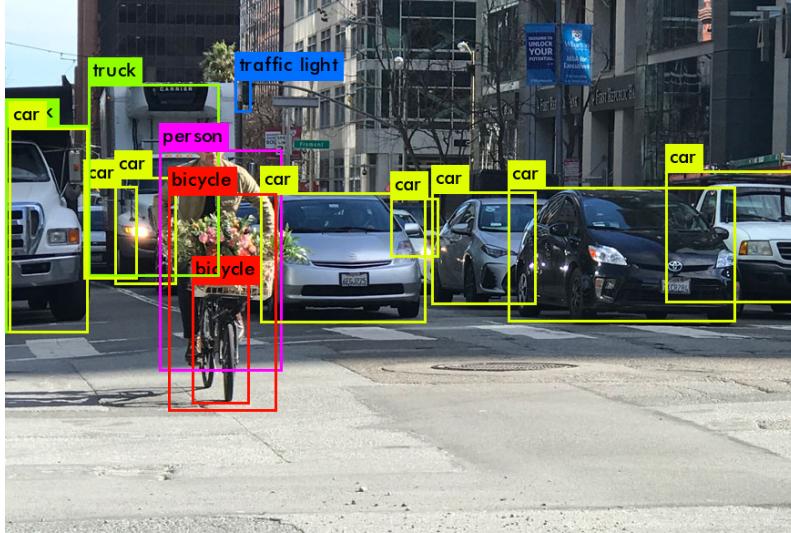


Figura 2.2: Object detection tramite l'algoritmo YOLO[14]

2.1.2 Segmentazione semantica

La segmentazione semantica è un’ulteriore tecnica utilizzata nell’ambito della computer vision. A differenza dell’object detection, la segmentazione semantica ha come obiettivo assegnare un’etichetta, rappresentante una classe, ad ogni pixel presente nell’immagine al fine di evidenziare le diverse regioni, le quali corrisponderanno ad oggetti o elementi specifici.

Per quanto riguarda le task di segmentazione semantica, negli anni sono stati tentati più approcci, utilizzando diversi modelli di reti neurali, dalle CNN alle reti neurali ricorrenti (RNN).

Le CNN in questo caso sono state usate per implementare modelli efficienti su task di estrazione delle feature, come Visual Geometry Group (VGG)[15] e ResNet[16]. VGG è un modello composto da 16 strati convoluzionali, in modo tale da riuscire ad espandere il campo recettivo dei neuroni, aumentando così il raggio di azione di ogni singolo neurone. ResNet invece ha la particolarità di possedere degli strati residuali, i quali permettono alla rete di riuscire a computare maggiormente in profondità, senza rischiare di incorrere nel problema della scomparsa del gradiente[17].

Le applicazioni della segmentazione semantica sono diverse e includono per esempio l’ambiente medicale, nel quale vengono utilizzate per identificare e analizzare l’anatomia nelle immagini mediche, migliorando così la diagnosi e il trattamento delle malattie. Nel contesto della guida autonoma, invece, vengono sfruttate per riconoscere e distinguere elementi lungo il percorso stradale, come ad esempio la segnaletica stradale e ostacoli di ogni tipo, grazie alla particolarità della segmentazione di evidenziare la forma di un oggetto, consentendo così ai modelli per la guida autonoma di prendere decisioni di conseguenza.



Figura 2.3: Segmentazione semantica applicata alla guida autonoma²

2.2 Pose estimation

La pose estimation, o stima della posa, è l'argomento alla base del progetto descritto in questa tesi ed è inoltre un'importante area della computer vision. Utilizzando questa tecnica ci concentriamo sull'identificazione e sulla comprensione della posizione di persone, o in generale esseri viventi, all'interno di un contenuto visuale, come un'immagine o un video.

Questa tecnologia viene utilizzata per risolvere diversi problemi, principalmente distinti in due tipologie, divisione effettuata in base alla dimensionalità del risultato: stima della posa 3D e stima della posa 2D. Un'altra divisione categorica dei problemi di pose estimation si basa sul numero di soggetti da rilevare, troviamo quindi *single-person estimation* e *multi-person estimation*.

In generale, ci sono vari ostacoli che devono essere considerati dalle tecnologie di stima della posa, infatti per esempio le immagini da cui estrarre la posa possono presentare configurazioni che rendono difficile l'identificazione, come movimenti particolari del soggetto o punti di vista sfavorevoli del dispositivo utilizzato per rilevare il soggetto[18], quest'ultimo particolarmente nel caso in cui per la rilevazione dell'immagine venga utilizzata una singola fotocamera.

Nelle seguenti sezioni verranno illustrate nel dettaglio le principali macrotecnicologie di stima della posa, concentrandoci su quelle basate su singolo soggetto utilizzando una sola fotocamera per l'acquisizione dell'input.

²<https://towardsai.net/p/1/machine-learning-7>

2.2.1 Pose estimation 2D

Per quanto riguarda i modelli che riescono a individuare la posa 2D di un singolo scheletro, i principali metodi che possiamo osservare sono quelli basati su *regressione lineare* e quelli basati su *mappe di calore* (*o heatmap*). Questi due approcci, come mostrato in Figura 2.4, si differenziano per il modo in cui riescono a identificare la posizione delle articolazioni del soggetto all'interno dell'immagine, il primo tramite operazioni di regressione applicate ai parametri di un modello scheletrico, il secondo invece tramite la predizione delle aree dell'immagine con maggiore probabilità di essere occupate da articolazioni.

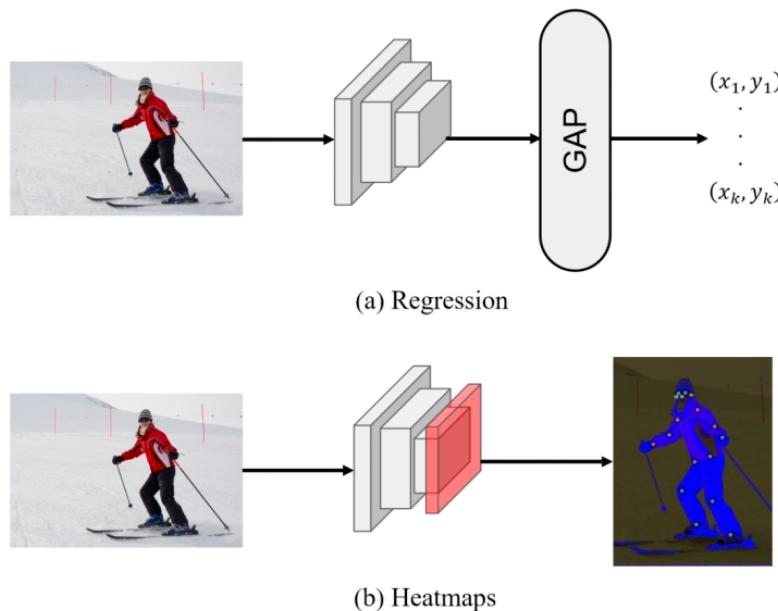


Figura 2.4: Diversi approcci per la stima della posa bidimensionale[19]

Metodi basati su regressione lineare

I metodi basati su regressione lineare rappresentano una tipologia di stima della posa 2D. Nei primi modelli sviluppati tramite questo metodo, venivano effettuate operazioni di regressione tra i parametri del modello e la posizione delle articolazioni, questo facendo riferimento a modelli di scheletro predefiniti. Questo approccio però presentava dei limiti, in quanto fare affidamento su modelli scheletrici rendeva dipendente la qualità delle predizioni dalla qualità degli scheletri utilizzati durante la fase di allenamento.

Con l'arrivo del deep learning, l'utilizzo di reti neurali convoluzionali assieme a tecniche di regressione ha permesso di ottenere risultati migliori rispetto agli approcci tradizionali appena citati, in quanto le CNN sono in grado di estrarre più facilmente informazioni rilevanti dalle immagini.

DeepPose Un esempio di applicazione di questo metodo, che utilizza regressione lineare e reti convoluzionali, è rappresentato da *DeepPose*[20]. Questo modello è in grado di identificare la posizione delle articolazioni, estraendo features dal contesto. L’architettura è composta da 7 layer in totale, di cui 5 convoluzionali e 2 completamente connessi. Una particolarità di questo modello è l’utilizzo di un metodo a cascata, nel quale la stessa rete neurale viene applicata più volte consecutivamente, ogni volta con un campo di ricerca più ristretto, migliorando così l’accuratezza delle predizioni.

Per ogni stadio s della cascata, dove s varia da 1 a $S = 3$, sono calcolati i parametri θ_s della rete neurale. ψ è la funzione di regressione della posa che riceve in input x e i parametri θ_s , calcolando $\psi(x; \theta_s)$.

Per rifinire una data posizione dell’articolazione y_i , viene considerata una *bounding box* (b_i) relativa all’articolazione, applicata alla sotto-immagine ottenuta nell’iterazione precedente:

$$b_i(y; \sigma) = (y_i, \sigma \text{diam}(y), \sigma \text{diam}(y))$$

dove $\text{diam}(y)$ è l’area in cui è stata identificata l’articolazione, considerando un offset arbitrario σ .

Spostandoci ora sulla definizione del primo stadio della cascata, nella fase $s = 1$ la computazione inizia con una bounding box b_0 , corrispondente all’intera immagine, ottenendo così una posa iniziale, definita da:

$$y_1 \leftarrow N^{-1}(\psi(N(x; b_0); \theta_1); b_0)$$

La prima fase di regressione, utilizzando la rete AlexNet[21], permette di ottenere una posa approssimativa, la quale verrà utilizzata come input per gli stadi successivi.

In ciascuna fase $s \geq 2$, per tutte le articolazioni verrà effettuata iterativamente una fase di perfezionamento della localizzazione, questo restringendo il campo sempre di più, relativamente al punto in cui viene identificata l’articolazione nell’iterazione precedente.

Utilizzando questo approccio, è stato possibile ottenere risultati migliori rispetto all’applicazione tradizionale tramite modelli scheletrici, sfruttando le potenzialità di estrarre features delle reti neurali convoluzionali.

Metodi basati su mappe di calore

Un metodo alternativo all’utilizzo di regressione lineare è quello basato su mappe di calore (o *heatmap*). Questo tipo di approccio si basa su due fasi di elaborazione, la prima consiste nell’identificare in quali aree dell’immagine è più probabile che si tro-

vino le articolazioni, la seconda invece nella localizzazione esatta delle articolazioni all'interno delle aree identificate.

Anche in questo caso l'utilizzo di reti convoluzionali è stato alla base di moltissime implementazioni basate su mappe di calore, la differenza principale tra le architetture allo stato dell'arte è stata quella di utilizzare in modo diverso le CNN, come ad esempio implementando una rete generativa avversaria (GAN)[22], oppure combinando l'utilizzo di moduli convoluzionali assieme a moduli basati su reti neurali ricorrenti[23].

PoseNet Il modello *PoseNet*[22] rappresenta un'implementazione di stima della posa basata su mappe di calore, nella quale viene utilizzato un sistema di generazione e discriminazione per predire in modo migliore la posizione delle articolazioni. Più in particolare, la rete può essere suddivisa in tre parti principali:

- **Generatore** Il generatore si occupa di generare le *heatmaps* delle articolazioni del soggetto, restituendo in totale 32 mappe di calore. La prima metà corrisponde ad ognuna delle 16 articolazioni individuate da questo modello, la seconda invece, indica la confidenza con cui ogni punto è stato identificato, utilizzando un valore compreso tra 0 e 1.
- **Discriminatore sulla posa** Questo componente ha l'obiettivo di valutare la qualità delle mappe generate, le pose stimate che non rispettano le regole di anatomia umana vengono penalizzate, così da non valutare le preidizioni completamente errate.
- **Discriminatore sulla confidenza** Questo discriminatore ha il compito di valutare la confidenza con cui ogni punto è stato identificato. Tale approccio ha permesso di valutare con maggior precisione i punti più difficili da identificare. Un esempio è dato dai punti coperti a causa della prospettiva da cui viene catturato il soggetto, che soprattutto nella prima parte della fase di training, vengono inevitabilmente predetti con un valore di confidenza minore. Rigettando sistematicamente queste generazioni, il modello riesce a migliorare la capacità di fare inferenza su questi punti.

Questo approccio ha permesso di ottenere risultati notevoli, come un valore percentuale di punti corretti individuati (PCK) pari al 92%, fornendo prestazioni migliori rispetto ad altri modelli basati su mappe di calore allo stato dell'arte.

2.2.2 Pose estimation 3D

Nel rilevamento della posa tridimensionale di un singolo soggetto possiamo osservare due possibili metodi: *skeleton-only* o *human mesh recovery (HMR)*[24]. I seguenti

metodi, vengono utilizzati per estrarre lo scheletro 3D del soggetto su immagini RGB ottenute da un singolo punto di vista.

Metodi a solo scheletro

2D to 3D lifting Questo tipo di approccio vuole sfruttare gli ottimi risultati ottenuti nella stima della posa bidimensionale, l'obiettivo infatti è quello di, dopo aver ottenuto le coordinate della posa su due assi, fare inferenza per ottenere le coordinate del terzo asse sui punti già a disposizione.

Come per il rilevamento della posa 2D, possono essere utilizzati diversi metodi per ottenere la rispettiva posa 3D, un lavoro che implementa una soluzione molto semplice è quello di Chen et al.[25], nella loro implementazione è stata utilizzata la transizione da 2D a 3D sfruttando la generazione di numerose coppie 2D-3D, in questo modo viene effettuata la predizione del valore del terzo asse imparando a selezionare la posa tridimensionale più simile a quella bidimensionale identificata.

Un approccio simile a quello descritto precedentemente nell'implementazione di PoseNet è stato utilizzato in RepNet[26], anche in questo caso sono state utilizzate le GAN per migliorare la qualità delle predizioni. Più in particolare, dopo aver generato una stima della posa 3D, la discriminazione può avvenire in due modi: utilizzando una *critic network*, nella quale viene valutata la qualità della posa tramite una funzione di perdita, oppure utilizzando una *reprojection network* che ha il compito di valutare la qualità della posa 3D generata, confrontando il risultato di una riproiezione da 3D a 2D, comparando la posa bidimensionale iniziale con quella rigenerata.

Direct estimation Un approccio alternativo è quello della *direct estimation*, in questa modalità la posa 3D viene stimata direttamente senza passare da rappresentazioni intermedie, questo si traduce nella maggior parte dei casi in un problema di regressione, simile a quello considerato per la posa 2D, differenziandosi solamente nella dimensionalità dell'output generato. Le operazioni di regressione vengono tipicamente effettuate utilizzando modelli di scheletro predefiniti, considerando le articolazioni oppure altre caratteristiche anatomiche del soggetto.

L'intuizione di Sun et al[27]. è stata quella di considerare la posa non solamente attraverso le articolazioni ma sfruttando anche le ossa. Questa applicazione è stata possibile grazie alla corrispondenza tra le strutture anatomiche in gioco, notiamo infatti che l'intersezione tra due ossa corrisponde ad un'articolazione. Utilizzando questo approccio, è possibile calcolare la funzione di perdita considerando più informazioni, non solo la posizione delle articolazioni, ma anche parametri altrettanto significativi come la lunghezza delle ossa.

Metodi basati sulla ricostruzione della superficie umana (HMR)

Un metodo differente rispetto alla stima della posa a solo scheletro è quello basato su *Human Mesh Recovery (HMR)*, questa tipologia di approccio prevede, per identificare il soggetto in uno spazio tridimensionale, la ricostruzione della sua superficie. Di seguito possiamo osservare due diverse classi di implementazioni, parametriche e non parametriche.



Figura 2.5: Applicazione di Human Mesh Recovery³

Metodi parametrici Tramite l'utilizzo di modelli parametrici è possibile stimare la superficie del soggetto, per questo lavoro viene tipicamente utilizzato lo *Skinned Multi-Person Linear model (SMPL)*[28]. Tramite SMPL è possibile rappresentare in modo accurato la forma del corpo umano, tenendo conto delle variazioni anatomiche ottenute dalle diverse pose, così da poter in seguito estrarre informazioni sulla posa del soggetto. Utilizzando modelli come SMPL riduciamo il problema di ottenimento della superficie in un problema di regressione, stimando i parametri del modello per adattarli alla posa del soggetto in input.

Un esempio di applicazione che coinvolge la ricostruzione della superficie umana è rappresentato da *BlazePose GHUM Holistic (BGH)*[29], questo modello viene utilizzato da MediaPipe Pose Landmarks Estimation, ovvero il framework scelto per la stima della posa in questo progetto. Il processo di acquisizione della posa avviene in due fasi distinte, la prima fase consiste nell'identificare la posa 3D del soggetto tramite un'operazione di 2D-to-3D lifting. Questa prima parte ha coinvolto l'utilizzo del modello BlazePose[30] per l'ottenimento della posa 2D. Per minimizzare l'errore di predizione sulla profondità della posa del soggetto invece, è stato utilizzato un

³<https://medium.com/snu-aiis-blog/expressive-3d-human-pose-and-shape-estimation-part-2-mesh-estimation-and-3d-rotational-pose-fa15c2149cc1>

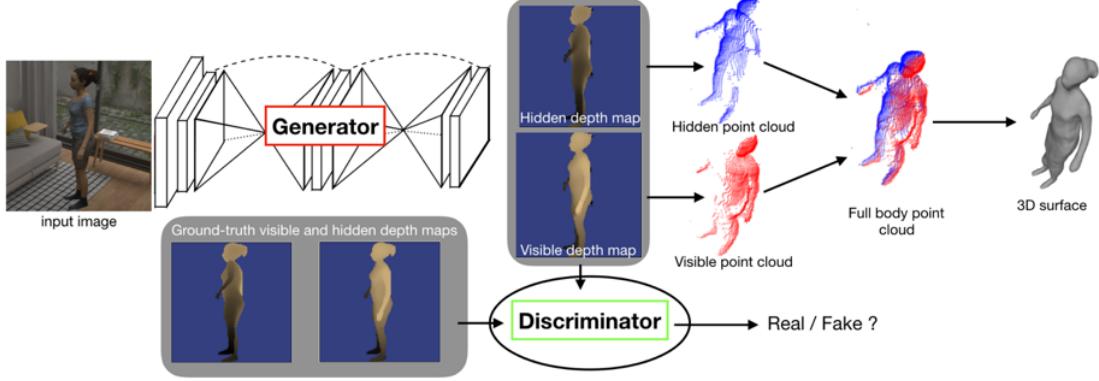


Figura 2.6: Architettura utilizzata in *Moulding Humans*[33]

dataset di pose tridimensionali annotate manualmente. In seguito, nella seconda fase viene utilizzato un modello parametrico per la ricostruzione della superficie del soggetto simile ad SMPL, GHUM[31], il quale permette di fare inferenza sulla forma del corpo del soggetto, ottenendo così pose più realistiche.

Una particolarità di BGH è l'attenzione posta sulla predizione degli arti superiori, infatti per ottenere una stima più accurata, compensando una mancanza del modello BlazePose, è stata effettuata una fase di predizione solo su di essi utilizzando un ulteriore modello, ottenendo in questo modo 21 punti di riferimento per la sola posa della mani.

Metodi non parametrici A differenza dell'approccio appena descritto, in questo caso la superficie del soggetto viene ricostruita senza utilizzare modelli di riferimento, questo permette di avere più flessibilità nella stima della posa, col rischio però di ottenere pose meno realistiche.

La flessibilità di questi modelli ha permesso di ottenere risultati migliori nella stima della posa di soggetti, comprendendo anche caratteristiche non considerate dai modelli parametrici, come ad esempio capelli o vestiti[32].

Nell'implementazione *Moulding Humans*[33] vengono stimate due superfici: una superficie “nascosta”, ovvero quella che da un’immagine bidimensionale come input risulta occlusa, e una superficie “visibile”. Tramite l’utilizzo di un’architettura GAN, queste superfici vengono generate per poi essere in seguito passate ad un discriminatore, il quale ha il compito di rigettare superfici non conformi o non abbastanza realistiche, così da allenare la rete ottenendo una migliore accuratezza.

2.3 Applicazioni nello sport

Come in ogni settore, lo sviluppo dell’intelligenza artificiale ha avuto un importante riscontro anche in ambito sportivo. Questo tipo di tecnologia è stata sfruttata al fine

di ottimizzare le performance degli atleti, migliorare strategie di gioco oppure per trovare correlazioni tra particolari movimenti e la possibilità di incorrere in infortuni, permettendo così lo sviluppo di un nuovo modo di interpretare e manipolare dati.

Le tipologie di informazioni analizzabili nel contesto dello sport sono numerose, tramite sensori indossabili, per esempio, è possibile monitorare la posizione di un atleta oppure raccogliere informazioni riguardanti la sua salute, come il battito cardiaco o la forza espressa durante le contrazioni muscolari. Invece, tramite telecamere, come quelle utilizzate per trasmettere in diretta una competizione, è possibile analizzare i movimenti degli atleti, sia presi singolarmente che in gruppo, al fine di analizzarne le strategie di gioco e valutarne l'efficacia.

2.3.1 Salute e sport

Un primo esempio, nel quale vengono applicate tecniche di deep learning su dati rilevati da sensori indossabili, è osservabile in questo studio di Wang et al.[34]. Sfruttando i sensori presenti negli smartwatch ad oggi più comuni, è stato possibile utilizzare una combinazione di CNN, per espandere le features, e un modello LSTM assieme ad un meccanismo di *self-attention* al fine di identificare correlazioni tra le sequenze di dati raccolte.

In questo progetto è stato utilizzato il dataset OPPORTUNITY[35], il quale raggruppa informazioni catturate da due tipi di sensori posti in determinate posizioni del corpo: accelerometri e sensori inerziali. Tramite questo set di informazioni è possibile allenare reti neurali con lo scopo di risolvere task di *Human Activity Recognition*.

Tramite questo approccio è stata raggiunta un'ottima accuratezza nella predizione dello stato di salute di una persona, riuscendo a fornire come output un report completo.

2.3.2 Strategie e sport di squadra

Numerose pubblicazioni hanno trattato l'applicazione dell'intelligenza artificiale agli sport di squadra, l'obiettivo di questi studi è prevalentemente la predizione dell'efficacia delle strategie adottate dai vari team. Ad esempio, in *Multi-Agent Deep-Learning Based Comparative Analysis of Team Sport Trajectories*[36] è stata valutata l'applicazione di reti neurali convoluzionali (CNN) e reti neurali ricorrenti (RNN) nel contesto del basket professionistico. In questo progetto è stato utilizzato un dataset contenente informazioni sulle dinamiche di gioco, come movimenti dei giocatori e del pallone, raccolti durante 600 partite del campionato NBA nella stagione 2015/2016.

L'obiettivo di questo lavoro era analizzare e predire l'efficacia delle strategie di gioco utilizzate durante le partite, classificando due problemi principali: identificare

strategie efficienti o non efficienti e prevedere azioni che avrebbero portato al conseguimento di un canestro oppure no. La metodologia di approccio proposta in questa ricerca ha coinvolto l'uso di modelli di reti neurali per elaborare e interpretare i dati di gioco, al fine di identificare pattern e correlazioni nel tempo che potessero predire il successo delle azioni di gioco.

2.3.3 Tecniche di allenamento e sport individuali

L'impiego dell'intelligenza artificiale (AI) nel contesto sportivo ha portato vantaggi significativi anche agli sport individuali. In generale, ha trovato varie applicazioni volte ad ottimizzare le metodologie di allenamento degli atleti. Ad esempio, ha reso possibile lo sviluppo di programmi di allenamento personalizzati e condotto analisi approfondite sulle prestazioni degli atleti, fornendo feedback mirati per migliorare ed ottimizzare le sessioni di allenamento.

Nella pubblicazione *Swimming style recognition and lap counting using a smart-watch and deep learning*[37], viene presentata un'applicazione dell'AI nel nuoto, questo sfruttando anche l'utilizzo di uno smartwatch. L'obiettivo era sviluppare un modello di rete neurale basato su CNN capace di riconoscere lo stile di nuoto eseguito da un atleta e di tenere conto del numero di vasche percorse. Questo risultato si rivela utile come supporto all'allenamento, consentendo, ad esempio, di monitorare il percorso di un nuotatore anche senza una presenza terza come quella di un allenatore. Questo sistema si è rivelato in grado di differenziare tra stili ed esercizi, riconoscendo in modo affidabile il momento in cui il nuotatore cambia stile, effettua una virata o si ferma per un certo periodo di tempo.

Un'altra importante applicazione nel campo del supporto all'allenamento tramite intelligenza artificiale, è rappresentata AI Coach[38]. In questo lavoro, l'obiettivo è fornire un sostegno nella valutazione e nella correzione dei movimenti eseguiti durante le routine di sci acrobatico. L'implementazione di questo sistema ha fatto uso della stima della posa dell'atleta durante l'esecuzione. Successivamente, la posa è stata confrontata con pose corrette, consentendo così una valutazione dettagliata dell'esecuzione e l'eventuale correzione dei movimenti. Tramite AI Coach è possibile rimpiazzare parzialmente un allenatore, ma soprattutto evitare all'atleta di dover guardare più volte un replay della sua prestazione per analizzarne la correttezza e per individuarne i difetti.

2.4 Ginnastica artistica e Calisthenics

La ginnastica artistica e il calisthenics sono due attività sportive, principalmente effettuate a corpo libero, che richiedono una combinazione di forza, flessibilità, coordinazione e propriocezione del corpo. In entrambi gli sport vengono eseguiti dei

movimenti di forza che richiedono molta precisione ed allenamenti mirati al fine di padroneggiarli.

Nella ginnastica artistica maschile per esempio, gli atleti eseguono una serie di routine sul pavimento o su vari attrezzi, come anelli, sbarra, parallele o cavallo con maniglie, i quali includono elementi di forza combinati ad elementi acrobatici e di equilibrio.

Nel calisthenics invece, disciplina particolarmente affine alla ginnastica, gli atleti si concentrano sull'utilizzo del proprio peso corporeo per eseguire combinazioni di movimenti dinamici e statici eseguiti in più varianti, come piegamenti, trazioni o verticali.

Entrambi gli sport, durante l'allenamento o durante le competizioni, richiedono un'analisi dettagliata dei movimenti e una comprensione approfondita della biomeccanica, sia per migliorare le prestazioni degli atleti e per prevenire infortuni, sia per valutarne la correttezza. In questi contesti possiamo considerare utile il supporto fornito dall'intelligenza artificiale, offrendo strumenti avanzati per analizzare e ottimizzare le tecniche di esecuzione.

Capitolo 3

Il progetto

Questo progetto di tesi si concentra sulla creazione di un modello di deep learning che sia in grado, all'interno di un'immagine, di identificare l'esercizio eseguito da un soggetto tramite l'utilizzo di coordinate spaziali, estratte tramite un modello di pose estimation.

Le quattro categorie di esercizi obiettivo della classificazione sono:

- *planche (o orizzontale)*
- *verticale*
- *front lever*
- *back lever*

Nella Figura 3.1 è possibile osservare esempi di immagini presenti nel dataset rappresentanti gli esercizi da riconoscere.

Dopo aver riconosciuto l'esercizio, tramite calcoli analitici applicati ai punti rilevati, si vuole valutare la correttezza dell'esercizio evidenziandone gli errori riconosciuti durante l'esecuzione.

3.1 Tecnologie utilizzate

Di seguito, un'introduzione alle principali tecnologie utilizzate per la realizzazione di questo progetto di tesi.

3.1.1 Python

Python[39], linguaggio introdotto da Guido van Rossum nel 1991, è un pilastro fondamentale della programmazione. Caratterizzato da tipizzazione dinamica e tipaggio forte, permette un approccio semplice alla programmazione fornendo una sintassi intuitiva.



Figura 3.1: Esempi di immagini presenti all'interno del dataset (in ordine dall'alto Front Lever, Planche, Back Lever e Verticale)

Python viene utilizzato per una vasta gamma di applicazioni, da soluzioni software fino ad utilizzi per compiti più avanzati di analisi e sviluppo. Una caratteristica distintiva di questo linguaggio è la sua libreria standard, la quale contiene numerosi moduli che permettono al programmatore di non dover implementare autonomamente funzioni di base.

Un ambito in cui questo linguaggio eccelle particolarmente è quello dell'intelligenza artificiale, infatti grazie a librerie apposite come *Tensorflow* e *PyTorch* è possibile creare, allenare e valutare modelli di reti neurali.

3.1.2 Tensorflow

TensorFlow¹ è una libreria open-source utilizzata nell'ambito dell'intelligenza artificiale e del calcolo numerico. Sviluppata da Google Brain², offre molteplici strumenti che la rendono una delle librerie più popolari per lo sviluppo e la distribuzione di modelli di apprendimento automatico su larga scala, una caratteristica importante che offre è la possibilità di utilizzare la GPU per velocizzare la parte di addestramento.

¹Tensorflow: <https://www.tensorflow.org/>

²Google Brain: <https://research.google/>

Per la creazione di architetture, TensorFlow mette a disposizione numerosi layer, come livelli densi, convoluzionali o LSTM, ma anche molte funzioni di attivazione, tra cui ReLU, Sigmoid o Softmax.

Riguardo l'addestramento di un modello, tramite Tensorflow è possibile utilizzare diversi ottimizzatori, come Adam, SGD o RMSprop, ma anche funzioni per il calcolo della perdita durante la fase di training, come la cross-entropy o la MSE.

La valutazione di un modello può essere effettuata tramite la funzione `evaluate`, la quale restituisce la perdita e l'accuratezza del modello sui dati di validazione.

Tramite Keras, una libreria di alto livello per la creazione di modelli di machine learning, viene estesa la funzionalità di TensorFlow, permettendo di creare modelli con pochi comandi e in modo più semplice.

3.1.3 Colab

Google Colab³ è un servizio basato su cloud che permette di scrivere ed eseguire codice Python tramite un browser web. Molto importante per le task di deep learning, è la memoria GPU che Colab mette a disposizione, tramite la quale è possibile allenare modelli, anche complessi, in tempi ragionevoli. Inoltre, Google Colab offre la possibilità di importare facilmente dataset, librerie e moduli di Python direttamente online, semplificando in questo modo il processo di sviluppo.

3.1.4 MediaPipe Pose Landmark Detection

MediaPipe⁴ è una libreria open-source sviluppata da Google che offre strumenti per la risoluzione di task di intelligenza artificiale, tra cui problemi di Computer Vision, Natural Language Processing (NLP) o per il processamento di file audio.

In questa tesi è stato utilizzato un modulo nell'ambito della visione artificiale, più in particolare per la *Pose Landmark Detection*, ovvero specifico per l'identificazione della posa umana e per il rilevamento delle sue articolazioni.

I punti rilevati tramite questo modello sono 33 (Figura 3.2) e le coordinate ritornate dal modello sono 3-dimensionali.

Per quanto riguarda la rilevazione della presenza di un soggetto, il modello di rete neurale utilizzato è molto simile a MobileNetV2[40], in questo caso ottimizzato per l'esecuzione in tempo reale, ma anche per essere compatibile con l'esecuzione su dispositivi mobili.

La stima della posa avviene invece tramite un modello variante di BlazePose[30], il quale utilizza GHUM[31], identificando le articolazioni tramite Human Mesh Recovery.

³Colab: <https://colab.google/>

⁴MediaPipe: <https://developers.google.com/mediapipe>

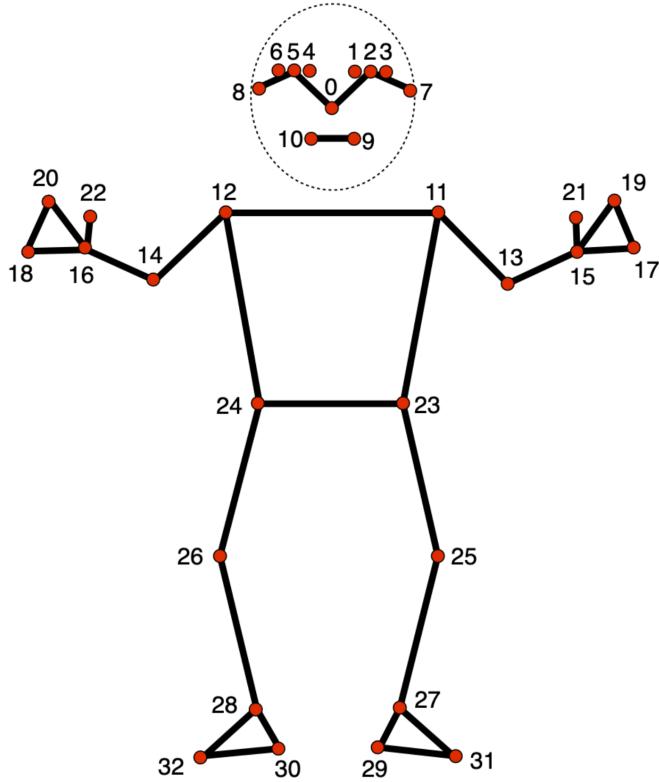


Figura 3.2: Punti rilevati dal modello di Pose Landmark Detection[30]

3.2 Creazione e pre-processamento del dataset

3.2.1 Raccolta dei dati

Il dataset utilizzato per l’addestramento del modello è stato creato tramite la raccolta di immagini sul web contenenti soggetti che eseguono gli esercizi specificati in precedenza.

Le immagini sono state raccolte in modo tale da avere un dataset il più possibile variegato e rappresentativo della realtà, sono presenti infatti sia esecuzioni di atleti professionisti, quindi con esecuzioni quasi impeccabili, sia esecuzioni di atleti amatoriali.

Inoltre, le immagini sono state raccolte da prospettive differenti rispetto al soggetto, in modo da rendere il modello più robusto e in grado di riconoscere gli esercizi anche in condizioni prospettiche non ottimali.

Questi dettagli hanno avuto un impatto significativo, considerando la dimensione ridotta del dataset. Raccogliere immagini da angolazioni diverse per ogni esercizio ha sicuramente aiutato a limitare il rischio di overfitting, contribuendo all’obiettivo di creare un modello in grado di generalizzare nel miglior modo possibile su nuovi dati.

In generale, per ogni esercizio sono state raccolte tra le 70 e le 73 immagini,

riuscendo così ad ottenere un dataset quasi perfettamente bilanciato, arrivando ad un totale di 280 immagini.

3.2.2 Pre-processing

Prima di procedere con la fase di addestramento, è stato necessario effettuare una parte di pre-processing sui dati raccolti, in modo da ottimizzarne la qualità e la compatibilità con il modello.

Un primo passaggio è stato quello di ridimensionare le immagini, in modo da focalizzare l'addestramento solamente sul soggetto che esegue l'esercizio, cercando di influenzare il modello il meno possibile con informazioni non rilevanti, come ad esempio altri possibili soggetti presenti nell'immagine.

Successivamente le immagini sono state passate al modello di pose detection, tramite il quale sono state ottenute le coordinate spaziali a tre dimensioni dei punti rilevati per ogni istanza del dataset. Le coordinate, utilizzando un metodo specifico contenuto nel modulo di stima della posa di MediaPipe, sono state raccolte già in forma normalizzata, in modo tale da avere valori compresi tra 0 e 1 per tutte e tre le coordinate, facendo sì che i punti rilevati siano indipendenti dalla grandezza dell'immagine.

Infine, le coordinate tridimensionali ottenute nella fase di pose detection sono state salvate in file di testo differenti per ciascuna immagine, ottenendo così il dataset finale utilizzato nella fase di addestramento dei modelli.

Per quanto riguarda l'etichettamento degli elementi del dataset, i file di testo sono stati raggruppati in cartelle diverse, ognuna rappresentante uno dei quattro esercizi da rilevare, automatizzando così il processo di labeling.

3.3 Specifiche dei modelli

Per la task di classificazione degli esercizi sono state realizzate due architetture differenti, così da poter effettuare un confronto e valutare quale dei due modelli sia più adatto per il problema affrontato.

3.3.1 Modello Dense

Il primo modello è basato su una rete completamente connessa, l'architettura finale ottenuta ha coinvolto un solo *hidden layer* composto da 8 neuroni, associato ad funzione di attivazione di tipo *ReLU* e ad un regolarizzatore del kernel di tipo L2. La scelta di utilizzare questo regolarizzatore è stata fatta in modo tale da prevenire situazioni spiacevoli di overfitting, riscontrate nei primi tentativi di addestramento della rete, nei quali non era presente alcun regolarizzatore.

Prima dello strato hidden è stato inserito un layer *Flatten*, nel quale le informazioni in input vengono modellate al fine di poter essere utilizzate come input nel layer denso. Nella parte finale della rete è presente un *output layer* completamente connesso associato ad una funzione di attivazione di tipo *softmax*, utilizzato per ottenere una distribuzione di probabilità per ciascuna delle quattro classi.

L'approccio appena descritto, con un solo strato intermedio, è stato selezionato dopo aver tentato l'utilizzo di più layer densi contemporaneamente a numeri più alti di neuroni (64, 128 o 256). Dopo diversi tentativi ho potuto riscontrare che, dovendo risolvere un problema piuttosto semplice per un modello deep, utilizzare un solo livello con un numero inferiore di neuroni mi ha permesso di poter allenare la rete per meno epoch ottenendo comunque risultati soddisfacenti.

Questa scelta ha influito anche sul numero di parametri, ottenendo un modello con una quantità considerevolmente minore di parametri, 428 in totale.

3.3.2 Modello CNN

Il secondo modello presenta invece un'architettura basata su una rete neurale convoluzionale (CNN). L'utilizzo di strati convoluzionali è giustificato dalle potenzialità da parte delle reti convoluzionali di apprendere pattern spaziali anche su informazioni non visive, come in questo caso, presentate come una sequenza di coordinate[41, 42, 43].

L'architettura finale comprende un solo strato intermedio, utilizzando un layer convoluzionale 2D, composto da 2 filtri, kernel di dimensione 3×3 , senza padding e con funzione di attivazione *ReLU*. Osservando la dimensione del kernel e la dimensione dell'input, si può notare come il layer convoluzionale sia stato utilizzato per apprendere pattern spaziali, in questo caso considerando combinazioni di 3 punti adiacenti tra loro per ogni singola posa analizzata.

Dopo questo strato sono stati aggiunti un layer di *Flatten* e un layer denso come *output layer*, con 4 neuroni e funzione di attivazione *softmax*, al fine di ottenere le probabilità relative alle quattro classi, corrispondenti agli esercizi da classificare.

Anche questo modello, non dovendo risolvere un problema particolarmente complesso ed essendo composto da un solo strato convoluzionale con pochi filtri, ha permesso di ottenere un modello composto in totale 408 parametri.

3.4 Addestramento dei modelli

3.4.1 Suddivisione del dataset

Per l'addestramento dei modelli è stato utilizzato il dataset precedentemente creato e pre-processato, considerando però non tutti e 33 i punti estratti dal modello di

pose detection, ma solamente 16 di questi (orecchie, spalle, gomiti, polsi, anche, ginocchia, caviglie e punte dei piedi per entrambi i lati), in modo tale da mantenere solamente le informazioni rilevanti per la classificazione degli esercizi, ovvero quelle la cui posizione potrebbe cambiare in base all'esercizio eseguito.

Il dataset è stato suddiviso in tre parti: 80% per l'addestramento del modello, il 10% per la validazione durante la fase di training ed il restante 10% come test set. Quest'ultima parte è stata utilizzata in seguito all'addestramento per valutare oggettivamente le prestazioni del modello su dati mai visti prima.

3.4.2 Fase di training

Durante la fase di addestramento sono state prese diverse decisioni, come la scelta dell'ottimizzatore, della funzione di perdita e degli iperparametri da utilizzare in ciascun modello.

La scelta dell'ottimizzatore è ricaduta su *Adam* per entrambi i modelli, ottimizzatore comunemente utilizzato e conosciuto per la sua versatilità nell'addestramento di reti neurali.

La funzione di loss è stata scelta in base al tipo di problema affrontato, è stata utilizzata infatti una funzione di tipo *sparse categorical crossentropy*, selezionata poiché adatta per problemi di classificazione multiclasse, come quello affrontato in questo progetto. Inoltre, questa funzione di loss è stata scelta dato che la sua implementazione in Tensorflow permette di gestire automaticamente la conversione delle etichette in forma *one-hot encoding*, gestendo in modo automatico la conversione delle etichette in forma numerica.

Modello Dense

Dopo aver allenato il primo modello sono stati ottenuti i risultati mostrati in Figura 3.3. Gli iperparametri utilizzati per l'addestramento sono i seguenti:

Iperparametro	Valore
Learning Rate (Adam)	0.002
Epoche	10
Batch Size	2

Tabella 3.1: Iperparametri e relativi valori utilizzati nella fase di addestramento del modello Dense

Modello CNN

Per quanto riguarda l'addestramento del modello basato su un layer intermedio convoluzionale, le performance ottenute durante la fase di training vengono mostrate in Figura 3.4. I parametri utilizzati durante la fase di training sono i seguenti:

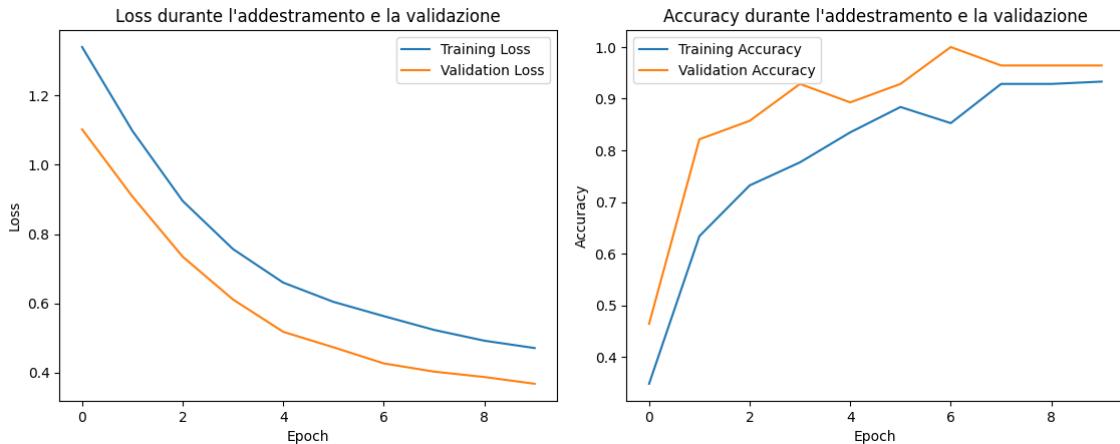


Figura 3.3: Andamento di loss e accuratezza durante l'addestramento del modello Dense

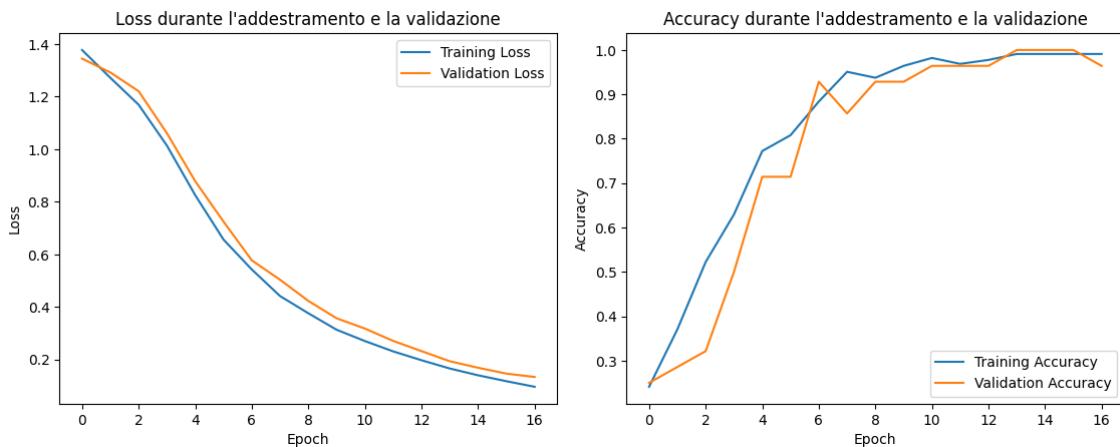


Figura 3.4: Andamento di loss e accuratezza durante l'addestramento del modello CNN

Iperparametro	Valore
Learning Rate (Adam)	0.002
Epoche	17
Batch Size	3

Tabella 3.2: Iperparametri e relativi valori utilizzati nella fase di addestramento del modello CNN

3.5 Considerazioni e confronto dei modelli

Analisi della fase di addestramento

Dai risultati ottenuti durante la fase di addestramento (Figura 3.3 e Figura 3.4) è possibile notare come la dimensione ridotta del dataset abbia influenzato le perfor-

mance dei modelli, permettendo di ottenere valori di accuratezza molto alti in tempi relativamente brevi.

Anche per questo motivo, con l'obiettivo di evitare overfitting, è stato scelto di addestrare entrambi i modelli per un numero di epoche limitato, in modo da evitare che i modelli memorizzassero troppe informazioni dai dati di addestramento, ottenendo di conseguenza una scarsa generalizzazione.

Analisi della fase di valutazione

Successivamente alla fase di addestramento di entrambi i modelli, è stato possibile confrontarli tra loro sui dati di test, in modo da valutare quale dei due sia più performante per il problema di classificazione risolto in questo progetto.

Come parametri di valutazione sono stati utilizzati la *loss*, l'*accuratezza* e l'*F1-score*.

Considerando anche il fatto di avere un dataset quantitativamente piccolo, la valutazione del modello è stata effettuata allenando il modello più volte su informazioni diverse del set, utilizzando seed diversi per la randomizzazione dello split del dataset, potendo utilizzare ogni volta come test set informazioni diverse.

Di seguito, una tabella riassuntiva dei risultati ottenuti durante la fase di valutazione sul test set. I valori riportati sono le medie dei risultati ottenuti utilizzando tre seed diversi per la suddivisione del dataset.

Modello	Loss	Accuratezza	F1-score
Dense	0.4692	0.9643	0.9642
CNN	0.1530	0.9762	0.9761

Tabella 3.3: Media dei risultati delle valutazioni dei modelli su test set diversi

Per fornire una maggiore comprensione dei risultati ottenuti, sono state calcolate le *confusion matrix* per entrambi i modelli (Figura 3.5), sempre utilizzando dati di test utilizzando tre seed diversi ad ogni iterazione per tre volte.

Da queste matrici di confusione è possibile notare come tutti e due i modelli abbiano ottenuto risultati molto simili, entrambi con pochi errori di classificazione nella quasi totalità degli esercizi, tranne che per il *Back Lever*, nella cui classificazione sono stati ottenuti un numero di falsi positivi maggiore rispetto agli altri esercizi.

Il modello Dense in particolare ha riscontrato un numero di falsi negativi superiore rispetto al modello CNN, scambiando spesso l'esercizio per un *Front Lever*. Questo comportamento può essere dovuto alla somiglianza tra le due posizioni, infatti l'unico tratto distintivo tra i due esercizi è la posizione del corpo, nel Front Lever rivolto verso l'alto e nel Back Lever rivolto verso il basso, posizione che può essere difficile da riconoscere per i modelli, specialmente considerando la prospettiva dell'immagine.

Nella Figura 3.6 è possibile osservare un esempio di esecuzione di un *Back Lever* riconosciuto come *Front Lever* dal modello Dense, assieme ad un esempio di *Front Lever* riconosciuto correttamente dallo stesso modello, entrambi con la stessa prospettiva.

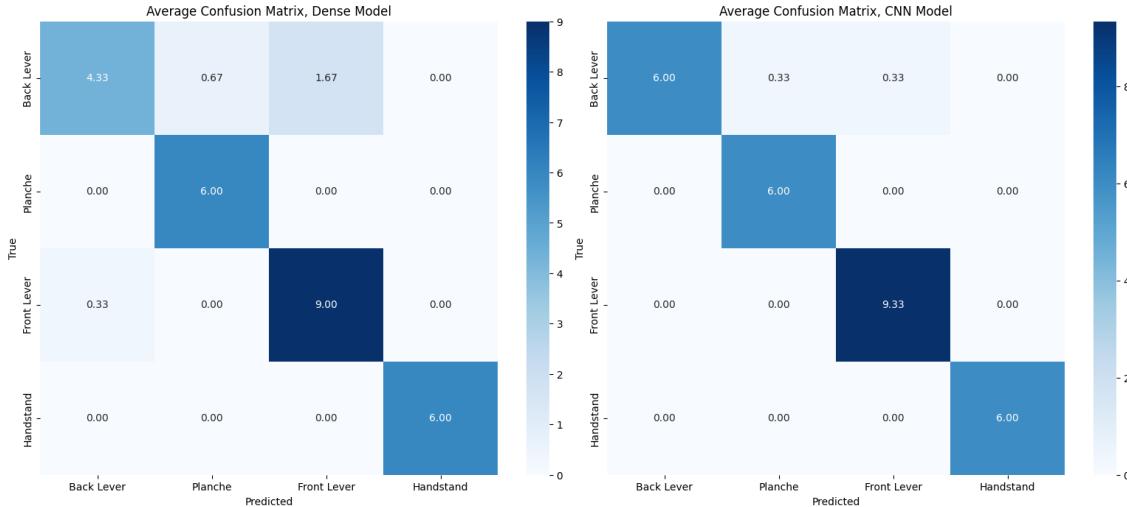


Figura 3.5: Confusion matrix ottenute dai modelli Dense e CNN sui dati di test

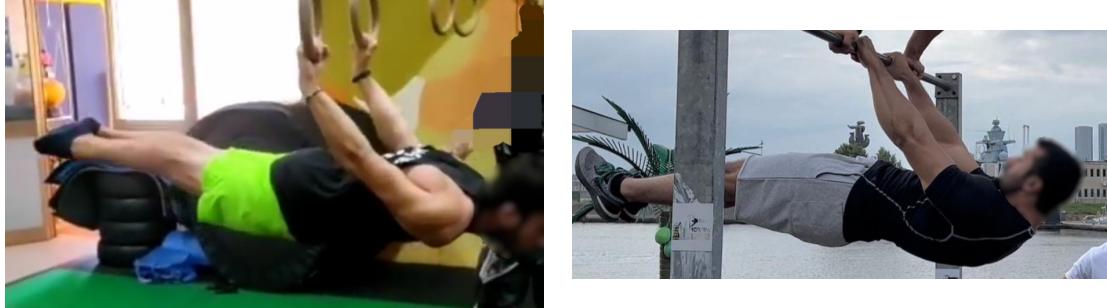


Figura 3.6: A sinistra, *Back Lever* riconosciuto come *Front Lever* dal modello Dense, a destra, *Front Lever* riconosciuto correttamente dal modello Dense

Anche osservando i risultati nella Tabella 3.3 si può notare come il modello basato su una rete neurale CNN abbia ottenuto risultati leggermente migliori rispetto al modello basato su strati densi.

Il valore di *loss*, che indica quanto il modello si discosti in media dalla soluzione corretta, è risultato essere notevolmente più basso per il modello CNN, con un valore di 0.1530, rispetto al modello Dense, il quale ha ottenuto un valore pari a 0.4692.

Per fornire una visione completa delle prestazioni dei modelli, è stato calcolato anche l'*F1-score*, valore che tiene conto di entrambi i valori di *precision* e *recall* calcolandone una media armonica, in modo tale da fornire un'ulteriore metrica utile alla valutazione delle prestazioni dei modelli.

Osservando i risultati ottenuti dal calcolo dell'*accuratezza* calcolata sul test set, in questo caso il modello convoluzionale ha ottenuto performance abbastanza simili

al modello completamente connesso, discostandosi di pochi punti percentuali, con un valore di 0.9762 per il modello CNN e 0.9643 per il modello Dense.

In generale, possiamo dire che entrambi i modelli hanno ottenuto risultati soddisfacenti per questo tipo di problema, se dovessimo scegliere il modello con prestazioni migliori la scelta ricadrebbe in questo caso sul modello basato su CNN, il quale, considerando una leggera maggiore accuratezza, ha ottenuto inoltre un valore di loss inferiore, indicando una maggiore robustezza in fase di classificazione e di conseguenza dimostrando di avere più sicurezza nelle predizioni rispetto al modello Dense.

3.6 Calcolo della correttezza di un esercizio

La seconda parte del progetto, successiva alla fase di identificazione dell'esercizio, è quella di valutazione della correttezza di un'esecuzione.

Per i criteri di valutazione è stato fatto affidamento al regolamento FIG (Federazione Internazionale di Ginnastica)⁵, il quale definisce delle linee guida comuni alla quasi totalità degli esercizi trattati in questo progetto.

Una scelta progettuale è stata quella di considerare valide anche posizioni riconosciute come “piccoli errori” nel regolamento FIG, ovvero posizioni che variano fino a 15° rispetto alla posizione ideale, considerando gli angoli formati dalle articolazioni principali (Figura 3.7). Un altro esempio di piccolo errore è quello data da posizioni inestetiche, come ad esempio le punte dei piedi non tirate. Questa decisione è stata presa in modo tale da evitare di incorrere in falsi positivi nel riconoscimento di errori, magari ottenuti da imprecisioni riconducibili al rilevamento della posa del soggetto, questo considerando che il modello di pose detection utilizzato possiede una percentuale di punti identificati correttamente (PCK) dell’84%[30], quindi non perfetta.

Mistake	Angular deviation	Deduction
Small	0° - 15°	0,1
Medium	>15° - 30°	0,3
Large	>30° - 45°	0,5
Large	>45°	0.5 + NR

Figura 3.7: Tabella di riferimento estratta dal regolamento che tratta la tolleranza riguardo deviazioni della linea del corpo e delle braccia

E’ importante sottolineare che, a differenza della fase di classificazione, in cui è possibile valutare immagini da qualunque prospettiva in cui si possa osservare

⁵Regolamento FIG - MAG CoP 2022-2024: <https://www.gymnastics.sport/site/rules/#2>

l'intero corpo del soggetto, per la fase di verifica della correttezza di un esercizio è necessario che l'immagine sia laterale rispetto al soggetto e parallela rispetto al terreno, in modo da poter analizzare correttamente, senza effettuare calcoli complessi che tengano conto della prospettiva, la posizione delle articolazioni.

Di seguito, i criteri base di valutazione utilizzati per la valutare la correttezza di un esercizio, effettuati su entrambi i lati del soggetto.

Corpo allineato

Per la valutazione del corpo in linea è stato necessario controllare l'angolo formato dai fianchi.

In questo caso non è stata presa in considerazione la Tabella 3.7, in quanto è presente un'ulteriore specifica riguardante la posizione dei fianchi, la quale è stata considerata come riferimento per questa parte di valutazione della correttezza dell'esercizio.

L'angolo ideale dovrebbe essere di 180° , consideriamo però valori $\geq 150^\circ$ come soglia, per restare consistenti nella scelta di valutare corretti anche errori considerati "piccoli" nel regolamento, in questo specifico caso tollerando angoli fino a 30° in meno rispetto all'ideale, come descritto nel regolamento (Figura 3.8).

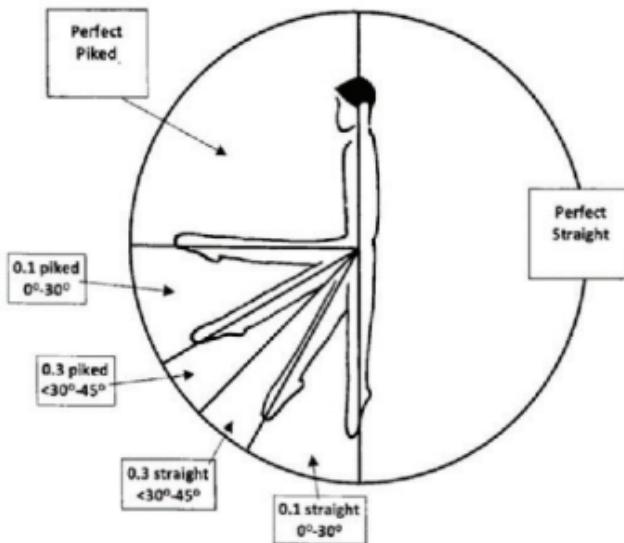


Figura 3.8: Criteri di valutazione estratti dal regolamento riguardo la linea del corpo, con relative soglie e penalità

Corpo in linea rispetto al terreno

Dopo aver controllato che il corpo fosse in linea, è stato necessario valutare che il corpo fosse:

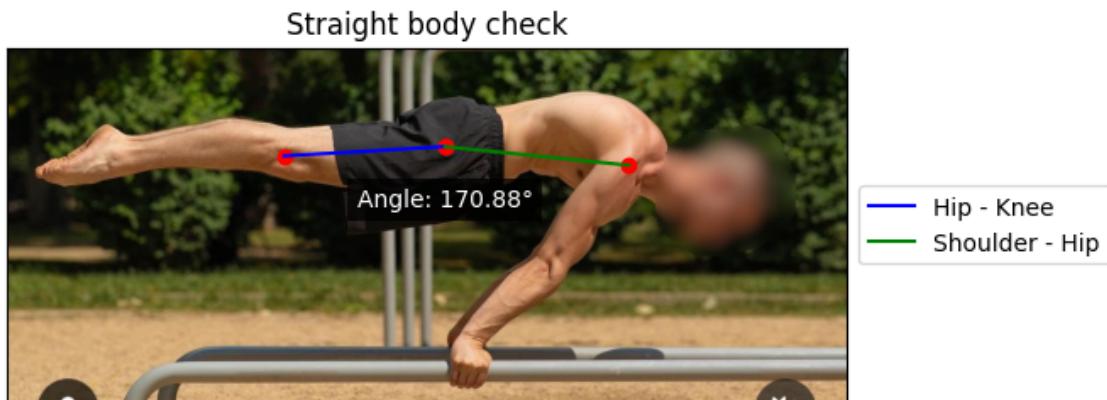


Figura 3.9: Esempio di valutazione del corpo allineato valutando l'angolo formato dai fianchi

- parallelo rispetto al terreno nelle posizioni di leva orizzontale come front lever, back lever e planche
- perpendicolare rispetto al terreno nella posizione di verticale

Nell'implementazione corrente, valutando la pendenza del corpo, è stato considerato un margine di incertezza di 15° , compatibilmente con il regolamento (Figura 3.7).

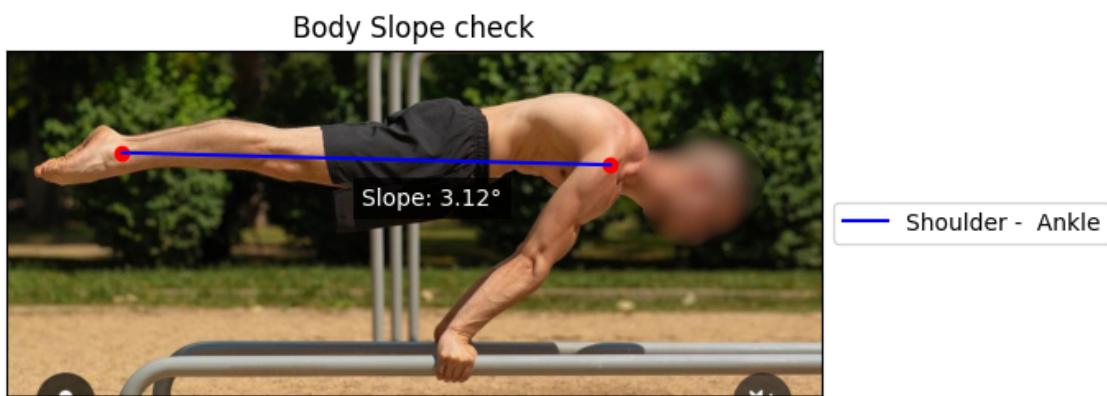


Figura 3.10: Esempio di valutazione della pendenza del corpo rispetto al terreno

Braccia tese

Un'ulteriore verifica necessaria è stata quella di calcolare l'angolo formato dal gomito, così da valutare che le braccia del soggetto fossero tese. Anche in questo caso l'angolo valutato dovrebbe essere idealmente di 180° , per restare consistenti alle precedenti valutazioni (Figura 3.7), ho considerato un'incertezza di 15° .

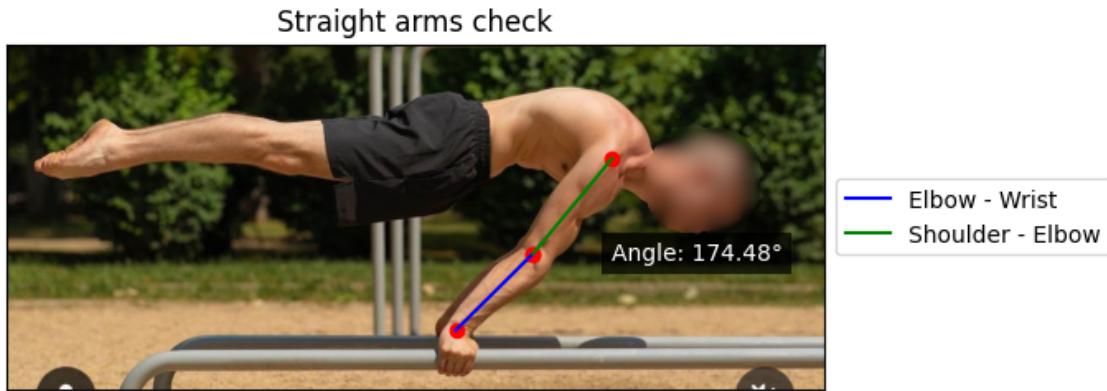


Figura 3.11: Esempio di valutazione delle braccia tese valutando l'angolo formato dal gomito

Punte dei piedi tirate

Infine, l'ultimo punto chiave analizzato per la valutazione della correttezza dell'esercizio è stata la posizione dei piedi. È quindi stato controllato che le punte dei piedi fossero tirate, considerando l'angolo di flessione plantare.

Brockett et al.[44] indicano che l'angolo di flessione plantare di una persona media si attesta tra i 40–55°. Pertanto, per valutare questo parametro rispetto all'angolo formato dalla caviglia, è stato considerato come corretto un valore di 145° (90° + 55°). Questo perché l'angolo di plantarflessione è considerato come l'angolo formato dal piede nella sua fase di estensione, oltre alla posizione normale in cui la caviglia forma un angolo retto.

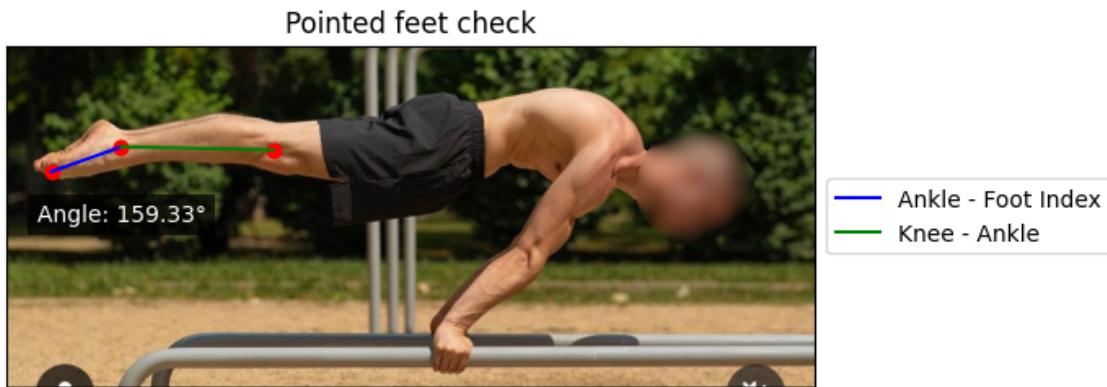


Figura 3.12: Esempio di valutazione delle punte dei piedi tirate valutando l'angolo di flessione plantare

Capitolo 4

Conclusioni

Tramite questo progetto è stato possibile sviluppare un semplice sistema di classificazione di esercizi statici a corpo libero a partire da un'immagine. Per farlo, è stato utilizzato un modello di pose estimation per estrarre i punti corrispondenti alle articolazioni del corpo umano, permettendo al modello classificatore di essere addestrato su queste informazioni. Oltre alla raccolta e al preprocessing delle informazioni che sono state utilizzate per costituire il dataset finale, sono stati sviluppati due modelli di deep learning differenti, il primo basato su uno strato convoluzionale e il secondo su strati completamente connessi. Dopo aver addestrato entrambi i modelli di classificazione, è stato possibile confrontarli, osservando in questo caso come il modello CNN abbia ottenuto risultati migliori rispetto al modello Dense.

4.1 Sviluppi futuri

Questo lavoro potrebbe essere ulteriormente approfondito e migliorato su diversi aspetti. Di seguito alcune possibili evoluzioni di questo progetto.

4.1.1 Allargamento del dataset e riconoscimento di nuovi elementi

Una miglioria che potrebbe essere apportata al modello di classificazione è quella data da un incremento del numero di informazioni all'interno del dataset, includendo sia nuovi esercizi, sia nuove immagini per gli esercizi considerati in questa prima versione. In entrambi i casi, potendo utilizzare più informazioni eterogenee durante la fase di training, il modello avrebbe sicuramente la possibilità di migliorare la sua capacità di generalizzazione, diventando più robusto e preciso, oltre che più adatto ad un utilizzo in contesti reali.

4.1.2 Classificazione tramite video e riconoscimento di elementi dinamici

Un altro possibile sviluppo di questo progetto potrebbe essere quello di classificare gli esercizi considerando non solo un’immagine statica, ma anche un video. Questa modifica potrebbe essere utile per poter valutare anche esercizi dinamici, tipicamente sempre presenti nelle performance eseguite durante le competizioni di ginnastica artistica o di calisthenics.

In un’ipotetica futura implementazione, sarebbe necessario modificare il modello di classificazione affinché possa ricevere in input un video. Inoltre, il classificatore dovrebbe essere in grado di trovare correlazioni tra la posizione delle articolazioni del soggetto analizzando frame consecutivi.

Questa modifica potrebbe essere realizzata tramite l’aggiunta di uno strato neurale ricorrente, come ad esempio un layer LSTM, sfruttando la capacità di quest’ultimo di memorizzare informazioni temporali, permettendo al modello di riconoscere le dinamiche presenti all’interno del video e di classificare l’esercizio in base a queste caratteristiche.

4.1.3 Adattamento della valutazione a prospettive diverse

Nell’attuale implementazione, la fase di valutazione della correttezza dell’esercizio viene eseguita analiticamente, con il limite che l’immagine debba essere scattata da una prospettiva il più possibile laterale.

Un’idea per migliorare questo aspetto potrebbe essere quella di sviluppare un modello di intelligenza artificiale anche per la valutazione dell’esercizio, riconoscendo un insieme limitato di possibili errori e valutando la correttezza dell’esercizio in base al numero di errori rilevati, implementando di fatto un sistema di *Anomaly Detection*.

Sfruttando la capacità delle reti neurali profonde, questo sistema potrebbe essere addestrato su un insieme di immagini che rappresentano gli errori più commessi durante l’esecuzione di ciascun esercizio classificabile. Utilizzando anche in questo caso un dataset di immagini scattate da diverse prospettive, il modello avrebbe la possibilità di adattarsi anche ad immagini non perfettamente laterali.

4.2 Applicazioni reali

Negli anni sono già stati sviluppati diversi sistemi predisposti alla valutazione tramite sistemi ICT delle performance nell’ambito della ginnastica artistica[3, 45]. Il problema di questi sistemi consiste principalmente nella poca portabilità, doven-

do settare in modo rigoroso la postazione di rilevamento e dalla necessità di avere sensori 3D per ottenere le informazioni sui movimenti del soggetto[46].

Tramite questo progetto si vuole semplificare e rendere più accessibile questo tipo di tecnologia, permettendo l'utilizzo di un sistema di classificazione e valutazione delle performance anche solo tramite uno smartphone e con meno restrizioni possibili sulla posizione di rilevamento.

Resta da valutare se un sistema di questo tipo possa essere utilizzato in ambito agonistico, dove la precisione e l'affidabilità delle informazioni sono elementi fondamentali nel processo di valutazione delle performance.

Considerando una versione più completa e accurata di questo sistema, migliorata come suggerito nei paragrafi precedenti, si potrebbe ottenere un valido strumento di supporto, sicuramente utile nella fase di preparazione ad una competizione, per ricevere feedback immediati sulla propria esecuzione.

Ringraziamenti

Desidero ringraziare il co-relatore di questa tesi, Giorgio Tsiotas, per il supporto e la disponibilità che mi ha fornito durante lo sviluppo di questo progetto, e il professore Andrea Aspertì per avermi dato l'opportunità di lavorare su questo argomento che è anche la mia passione fuori dall'ambito accademico.

Elenco delle figure

2.1	Schema del processo di elaborazione tipico della computer vision[4]	9
2.2	Object detection tramite l'algoritmo YOLO[14]	11
2.3	Segmentazione semantica applicata alla guida autonoma ²	12
2.4	Diversi approcci per la stima della posa bidimensionale[19]	13
2.5	Applicazione di Human Mesh Recovery ³	17
2.6	Architettura utilizzata in <i>Moulding Humans</i> [33]	18
3.1	Esempi di immagini presenti all'interno del dataset (in ordine dall'alto Front Lever, Planche, Back Lever e Verticale)	24
3.2	Punti rilevati dal modello di Pose Landmark Detection[30]	26
3.3	Andamento di loss e accuratezza durante l'addestramento del modello Dense	30
3.4	Andamento di loss e accuratezza durante l'addestramento del modello CNN	30
3.5	Confusion matrix ottenute dai modelli Dense e CNN sui dati di test .	32
3.6	A sinistra, <i>Back Lever</i> riconosciuto come <i>Front Lever</i> dal modello Dense, a destra, <i>Front Lever</i> riconosciuto correttamente dal modello Dense	32
3.7	Tabella di riferimento estratta dal regolamento che tratta la tolleranza riguardo deviazioni della linea del corpo e delle braccia	33
3.8	Criteri di valutazione estratti dal regolamento riguardo la linea del corpo, con relative soglie e penalità	34
3.9	Esempio di valutazione del corpo allineato valutando l'angolo formato dai fianchi	35
3.10	Esempio di valutazione della pendenza del corpo rispetto al terreno .	35
3.11	Esempio di valutazione delle braccia tese valutando l'angolo formato dal gomito	36
3.12	Esempio di valutazione delle punte dei piedi tirate valutando l'angolo di flessione plantare	36

Bibliografia

- [1] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” *Computational Intelligence and Neuroscience*, vol. 2018, p. 7068349, Feb 2018.
- [2] Z. Zhao, W. Chai, S. Hao, W. Hu, G. Wang, S. Cao, M. Song, J.-N. Hwang, and G. Wang, “A survey of deep learning in sports applications: Perception, comprehension, and decision,” 2023.
- [3] H. Fujiwara and K. Ito, “Ict-based judging support system for artistic gymnastics and intended new world created through 3d sensing technology,” *Fujitsu scientific & technical journal*, vol. 54, no. 4, pp. 66–72, 2018.
- [4] S. Paneru and I. Jeelani, “Computer vision applications in construction: Current state, opportunities and challenges,” *Automation in Construction*, vol. 132, p. 103940, 2021.
- [5] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.
- [6] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [7] P. Bharati and A. Pramanik, “Deep learning techniques—r-cnn to mask r-cnn: A survey,” in *Computational Intelligence in Pattern Recognition* (A. K. Das, J. Nayak, B. Naik, S. K. Pati, and D. Pelusi, eds.), (Singapore), pp. 657–668, Springer Singapore, 2020.
- [8] D. A. Pisner and D. M. Schnyer, “Chapter 6 - support vector machine,” in *Machine Learning* (A. Mechelli and S. Vieira, eds.), pp. 101–121, Academic Press, 2020.
- [9] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

- [10] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015.
- [11] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, “A review of yolo algorithm developments,” *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022. The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 and 2021): Developing Global Digital Economy after COVID-19.
- [12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21–37, Springer, 2016.
- [13] P. Rajeshwari, P. Abhishek, and P. S. . T. Vinod, “Object detection: An overview,” *International Journal of Trend in Scientific Research and Development*, vol. Volume-3, p. 1663–1665, Apr. 2019.
- [14] K. R. Velasco, “Yolo (you only look once),” Jun 2019.
- [15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015.
- [16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.
- [17] F. Lateef and Y. Ruichek, “Survey on semantic segmentation using deep learning techniques,” *Neurocomputing*, vol. 338, pp. 321–348, 2019.
- [18] D. Zhang, Y. Wu, M. Guo, and Y. Chen, “Deep learning methods for 3d human pose estimation under different supervision paradigms: A survey,” *Electronics*, vol. 10, no. 18, 2021.
- [19] X. Zhang and Q. Zhou, “Repnet: A lightweight human pose regression network based on re-parameterization,” *Applied Sciences*, vol. 13, no. 16, 2023.
- [20] A. Toshev and C. Szegedy, “Deeppose: Human pose estimation via deep neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
- [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.

- [22] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang, “Adversarial posenet: A structure-aware convolutional network for human pose estimation,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [23] V. Belagiannis and A. Zisserman, “Recurrent human pose estimation,” in *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*, pp. 468–475, IEEE, 2017.
- [24] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, “Deep learning-based human pose estimation: A survey,” *ACM Comput. Surv.*, vol. 56, aug 2023.
- [25] C.-H. Chen and D. Ramanan, “3d human pose estimation = 2d pose estimation + matching,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [26] B. Wandt and B. Rosenhahn, “Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [27] X. Sun, J. Shang, S. Liang, and Y. Wei, “Compositional human pose regression,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [28] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, “Smpl: a skinned multi-person linear model,” *ACM Trans. Graph.*, vol. 34, oct 2015.
- [29] I. Grishchenko, V. Bazarevsky, A. Zanfir, E. G. Bazavan, M. Zanfir, R. Yee, K. Raveendran, M. Zhdanovich, M. Grundmann, and C. Sminchisescu, “Blazepose ghum holistic: Real-time 3d human landmarks and pose estimation,” *arXiv preprint arXiv:2206.11678*, 2022.
- [30] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, “Blazepose: On-device real-time body pose tracking,” *arXiv preprint arXiv:2006.10204*, 2020.
- [31] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, “Ghum & ghuml: Generative 3d human shape and articulated pose models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6184–6193, 2020.
- [32] Y. Liu, C. Qiu, and Z. Zhang, “Deep learning for 3d human pose estimation and mesh recovery: A survey,” 2024.

- [33] V. Gabeur, J.-S. Franco, X. Martin, C. Schmid, and G. Rogez, “Moulding humans: Non-parametric 3d human shape estimation from single images,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- [34] T. Y. Wang, J. Cui, and Y. Fan, “A wearable-based sports health monitoring system using cnn and lstm with self-attentions,” *PLOS ONE*, vol. 18, pp. 1–14, 10 2023.
- [35] D. Roggen, A. Calatroni, L.-V. Nguyen-Dinh, R. Chavarriaga, and H. Sagha, “OPPORTUNITY Activity Recognition.” UCI Machine Learning Repository, 2012.
- [36] Z. Ziyi, R. Bunker, K. Takeda, and K. Fujii, “Multi-agent deep-learning based comparative analysis of team sport trajectories,” *IEEE Access*, vol. 11, pp. 43305–43315, 2023.
- [37] G. Brunner, D. Melnyk, B. Sigfússon, and R. Wattenhofer, “Swimming style recognition and lap counting using a smartwatch and deep learning,” in *Proceedings of the 2019 ACM International Symposium on Wearable Computers*, ISWC ’19, (New York, NY, USA), p. 23–31, Association for Computing Machinery, 2019.
- [38] J. Wang, K. Qiu, H. Peng, J. Fu, and J. Zhu, “Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance,” in *Proceedings of the 27th ACM International Conference on Multimedia*, MM ’19, (New York, NY, USA), p. 374–382, Association for Computing Machinery, 2019.
- [39] A. Nagpal and G. Gabrani, “Python for data analytics, scientific and technical applications,” in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pp. 140–145, 2019.
- [40] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.
- [41] W. Wu, Z. Qi, and L. Fuxin, “Pointconv: Deep convolutional networks on 3d point clouds,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [42] J. Mao, X. Wang, and H. Li, “Interpolated convolutional networks for 3d point cloud understanding,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

- [43] L. Wang, Y. Huang, J. Shan, and L. He, “Msnet: Multi-scale convolutional network for point cloud classification,” *Remote Sensing*, vol. 10, no. 4, 2018.
- [44] C. L. Brockett and G. J. Chapman, “Biomechanics of the ankle,” *Orthopaedics and trauma*, vol. 30, no. 3, pp. 232–238, 2016.
- [45] B. Reily, H. Zhang, and W. Hoff, “Real-time gymnast detection and performance analysis with a portable 3d camera,” *Computer Vision and Image Understanding*, vol. 159, pp. 154–163, 2017. Computer Vision in Sports.
- [46] A. Ejiri, K. Iida, H. Tomimori, Y. Ikai, S. Yamao, K. Teduka, K. Yanai, and M. Nishikawa, “3d sensing of gymnastics competition using mems mirror laser sensor,” in *2021 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 1175–1180, 2021.