

POLITECNICO DI TORINO

Laurea Magistrale in ingegneria biomedica



Tesi magistrale

Soluzioni digitali a supporto della canoa slalom

Relatori

Prof. Luca ULRICH

Prof. Giorgia MARULLO

Candidato

Giulio MAZZONI

03 2025

Indice

1	Abstract	I
2	Background	1
2.1	Canoa Slalom	1
2.2	Altri sport	2
2.3	Difficoltà nella canoa	3
2.4	Stato dell'arte	4
3	Metodi	7
3.0.1	Addestramento della rete neurale	9
3.0.2	Database fotografico	9
3.0.3	Processo di marcatura	11
3.0.4	Parametri di addestramento e risultati	13
3.1	Regia	16
3.1.1	Design UI	18
3.1.2	Gestione di flussi video multipli	19
3.1.3	Logiche interne e risoluzione problematiche	21
3.2	Analisi della traiettoria	24
3.2.1	Correzione della distorsione	27
3.2.2	Identificazione parametri intrinseci	29
3.2.3	Identificazione parametri estrinseci	31
3.2.4	Stima della matrice P	33
3.2.5	Decomposizione della matrice p	34
4	Risultati	37
4.1	Future work	42
4.1.1	Identificazione atleti	42
4.1.2	Tracciamento multiplo e upgrade camere	42
4.1.3	Utilizzo droni	43
4.1.4	Lavoro al vogatore	43
4.1.5	Ampliamento ad altri sport	44

5 Conclusioni	45
Elenco delle figure	48
Bibliografia	49

Capitolo 1

Abstract

L'utilizzo dell'Intelligenza Artificiale negli ultimi anni è aumentato significativamente grazie agli avanzamenti tecnologici che permettono la costruzione di computer sempre più potenti. Diventando alla portata di tutti, grazie anche al rilascio delle prime versioni di ChatGPT e di altri progetti indipendenti open-source, ci si è iniziati ad interrogare in quali e quanti ambiti potesse essere utilizzata. Uno dei principali, è risultato l'ambito sportivo. Sfruttando la sua capacità di analisi, calcolo ed elaborazione di molto superiori a quelle di un essere umano, sono nati i primi database costruiti ad hoc per l'addestramento di IA.

Fornendo in input una serie di immagini o video è possibile effettuare object detection, tecnica principale utilizzata in questo progetto di tesi. Marcando il soggetto di interesse, si può insegnare all'IA a riconoscerlo e a seguirlo durante una registrazione o un video in diretta. Molto diffuso negli sport principali come calcio e tennis, questo metodo viene espanso poco ad altri sport.

Nei successivi capitoli si vedrà come la tecnica dell'object detection è stata sfruttata per analizzare le prestazioni di atleti di Canoa Slalom, grazie alla gentile partecipazione dell'Ivrea Canoa Club, centro di allenamento tra i migliori nazionali e internazionali, che ha messo a disposizione il suo sistema di telecamere per la costruzione di un database adeguato. Lungo il percorso di discesa sono infatti presenti 5 camere che inquadrano tutto il canale da diverse angolazioni, con le quali è possibile seguire l'intera discesa degli atleti. Partendo da uno scopo prettamente registico, ci si è poi spostati verso l'analisi della traiettoria.

La rete neurale è capace di riconoscere quando, in una determinata inquadratura, è presente o meno l'atleta e, di conseguenza, seguirlo lungo il canale. Questo avviene con un controllo automatico della telecamera da mostrare ad un osservatore, a fronte dell'analisi contemporanea di tutti e 5 i flussi video. Per

quanto riguarda l'analisi della traiettoria, importante elemento nella prestazione finale dell'atleta, si parte identificando il centro del bounding box contenente la canoa e ne si traccia il percorso. In un secondo momento, vengono estratte le coordinate dei punti, espresse in termini di pixel. Tramite l'utilizzo dei parametri intrinseci delle telecamere e di conversioni geometriche, risulta possibile risalire alle coordinate reali, intese come coordinate spaziali sul piano del canale. Un lavoro futuro, aggiungendo più flussi video e integrando telecamere di profondità, porterà ad un incremento nella precisione delle rilevazioni e nelle informazioni a disposizione.

Per valutare l'effettiva prestazione dell'atleta è stato infine implementato un valore indicativo rappresentante la differenza tra una traiettoria e l'altra. Esso è dato dalla distanza tra i punti della nuova traiettoria rispetto a quella presa come riferimento, andando a confrontare questa distanza con delle soglie considerate plausibile. In uscita avremo un valore percentuale dove lo 0 rappresenta una traiettoria identica, mentre il 100 ne rappresenta una completamente diversa. Logicamente la traiettoria migliore risulta essere quella per cui si è impiegato meno tempo ad effettuare la discesa.

Capitolo 2

Background

La canoa slalom è uno sport particolare che unisce la necessità di preparazione atletica e mentale in egual modo. Ogni atleta deve essere capace di adattarsi alle condizioni che gli si presentano davanti e prendere decisioni precise in frazioni di secondo, valutando le caratteristiche del percorso di discesa e cercando di ottenere un vantaggio sfruttando la corrente ed evitando quanto più possibile penalità. Presente stabilmente dalle Olimpiadi del 1992, è una disciplina che rientra negli sport acquatici, più specificatamente, sport in acque mosse.

2.1 Canoa Slalom

Obiettivo di tale sport è completare un percorso lungo un canale, naturale o ricavato artificialmente, attraverso un susseguirsi di porte mantenute sospese a pelo d'acqua. La lunghezza del percorso è tra i 200 e i 300 metri su cui vengono poste tra le 18 e le 25 porte, composte da due pali verticali.

Queste ultime possono avere due colorazioni diverse, indicanti la direzione in cui vanno percorse: le verdi e bianche vanno attraversate nel verso della corrente, mentre le rosse e bianche vanno attraversate contro corrente, incrementando la difficoltà del percorso, i tempi di percorrenza e il dispendio fisico dell'atleta.

Le maggiori penalità assegnate riguardano proprio l'approccio alle porte. Nel caso in cui un atleta tocchi con pagaia, canoa o qualsiasi parte del corpo uno dei pali delle porte, gli verrà assegnata una penalità sul tempo complessivo di 2 secondi. Invece, nel caso in cui l'atleta attraversi una porta nel verso sbagliato o la salti completamente, la penalità sarà di 50 secondi, andando a minare quasi sicuramente la possibilità di ottenere un buon risultato.

Il tempo finale complessivo viene calcolato sommando il tempo di percorrenza alle eventuali penalità accumulate durante la discesa. La competizione è vinta da

chi possiede il tempo più basso. All'interno della disciplina della canoa slalom sono presenti molteplici categorie, divise in base al tipo di imbarcazione utilizzata.

Nella specialità K1, l'atleta è seduto nella canoa e usa una pagaia a doppia pala. Contrariamente, nella categoria C1, l'atleta si trova in ginocchio nella canoa e usa una pagaia a singola pala, comportando quindi un incremento dal punto di vista del dispendio fisico e della difficoltà di gestione della canoa. In ultimo, esiste la categoria C2 in cui 2 atleti si trovano nella stessa canoa e devono essere coordinati al meglio per garantire un risultato ottimale.

La C2 nelle moderne competizioni è molto meno comune. Si è quindi pensato di andare a fornire un supporto tecnologico agli atleti di categoria K1 e in parte anche C1, che, rispetto agli sport maggiori, ricevono meno attenzione.

2.2 Altri sport

Facendo alcuni esempi, vediamo come l'introduzione della VAR (Video Assistant Referee) nel calcio sia stata una rivoluzione. Questo sistema permette agli arbitri di rivedere alcune situazioni dubbiose dal punto di vista di multiple telecamere. Esso quindi viene utilizzato soprattutto in situazioni cruciali come, falli da rigore, espulsioni dirette o contestazioni.

Nonostante questo supporto scongiuri spesso errori arbitrali, il suo utilizzo risulta ancora contestato a causa dei tempi di revisione e interpretazione delle immagini fornite. Altra soluzione implementata nei tempi moderni è la Goal-Line Technology. Grazie a sensori e telecamere posti sulla porta, permettere di verificare con assoluta certezza se il pallone ha superato completamente la linea di porta. In questo modo, è stato possibile eliminare ogni dubbio nei casi di gol fantasma o indecisione arbitrale.

Spostando l'attenzione sul tennis, si vede come il sistema Hawk-Eye sia stato ampiamente adottato in tutte le competizioni più importanti. Si tratta di un sistema di telecamere atte al tracciamento della traiettoria della palla e a fornirne una rappresentazione virtuale durante l'impatto con il campo. Questo strumento è principalmente utilizzato per capire se una palla, al momento del contatto, si trova in campo o fuori campo, nei momenti di indecisione. Inoltre, è facoltà degli atleti richiedere un'analisi per verificare una decisione arbitrale, incrementando così la giustizia e correttezza durante le partite.

In ultimo, prendendo in analisi il basket, troviamo l'Instant Replay e l'utilizzo di realtà e intelligenza artificiale durante gli allenamenti. L'Instant Replay è una tecnologia principalmente usata sempre in ambito arbitrale, che rende disponibili immediatamente filmati in alta definizione dell'azione appena compiuta. Così

facendo, verifiche di tiro da tre punti o determinazioni di falli antisportivi, possono essere svolte istantaneamente.

Durante gli allenamenti poi, si è visto come l'utilizzo di ambienti virtuali abbia preso piede. Alcune squadre infatti, adottano simulatori VR per ricreare situazioni di gioco realistiche, permettendo agli atleti di allenarsi nella lettura di gioco e capacità decisionali sotto pressione, senza per forza incorrere in un affaticamento fisico. Affaticamento che, durante le partite, viene anche tenuto sotto controllo tramite tecnologie di monitoraggio per scongiurare infortuni e ottimizzare al meglio i minuti di gioco e recupero degli atleti.

2.3 Difficoltà nella canoa

E' chiaro come, negli sport sopracitati, ma non solo, la maggior parte delle applicazioni tecnologiche sia mirata ad un miglioramento della qualità dell'arbitraggio e, secondariamente, a fornire un supporto nell'allenamento. L'applicazione di tecnologie simili quindi, porterebbe anche nella canoa una maggiore precisione nelle competizioni e un maggiore coinvolgimento da parte di pubblico, giudici e atleti.

Uno dei problemi principali di questo sport risulta essere la valutazione delle penalità. Ad oggi, tutto si basa sull'attenzione dei giudici, la cui valutazione deriva esclusivamente da un'attenta osservazione degli atleti durante la discesa, ma come è facile immaginare, questo processo è soggetto ad errori umani e cattive interpretazioni.

Introducendo la possibilità di avere filmati di alta qualità subito disponibili, porterebbe a facilitare il compito dei giudici e a migliorare la correttezza nel processo di assegnazione di penalità, che, in caso contrario, potrebbero, in alcuni casi, compromettere un'intera competizione.

Similmente, un'analisi di traiettoria porterebbe vantaggi sia in gara che in allenamento. Nel primo caso si potrebbe avere un'ulteriore conferma del corretto passaggio attraverso una porta. Nel secondo, essendo la traiettoria l'elemento fondamentale che garantisce un buon tempo totale, si potrebbero andare a svolgere allenamenti mirati a limare le maggiori imperfezioni.

Anche l'impiego di sensoristica, con conseguente raccolta di dati su performance fisiche, darebbero un aiuto nel processo di allenamento e di preparazione ad una gara. Questo permetterebbe di avere informazioni su angoli di pagaiata, equilibrio velocità e accelerazione, permettendo agli allenatori di correggere efficientemente eventuali errori compiuti dagli atleti. Infine, applicando modelli di intelligenza artificiale, si potrebbe arrivare alla previsione delle migliori strategie di approccio a determinate rapide o configurazione di porte.

2.4 Stato dell'arte

Si è quindi passati all'analisi dei possibili mezzi con cui sopperire ad alcune delle precedenti mancanze. Inizialmente si è pensato all'utilizzo di sensori inerziali. Questi sistemi, chiamati IMU (Inertial Measurement Unit), fornendo dati sull'orientamento, accelerazione e velocità, avrebbero potuto dare un buon supporto durante la fase di allenamento. Il problema però risiede innanzitutto nel fatto che essi non diano informazioni sulla posizione globale dell'atleta, di conseguenza non sono adatti per un'analisi della traiettoria. Allo stesso tempo, sono più adatti a sport di squadra, rispetto a sport singoli.

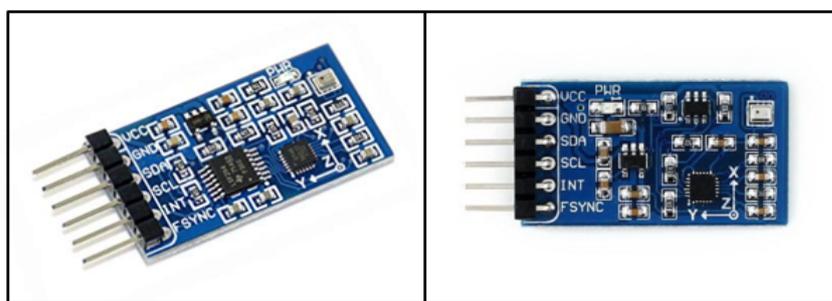


Figura 2.1: Esempi di IMUs

Negli sport di squadra è importante la posizione dei giocatori rispetto agli altri e in generale il movimento relativo degli stessi. Al contrario negli sport singoli, la tecnica fa da padrona, comportando una mole di dati molto maggiore, che i sensori non sono ancora efficienti ad elaborare e fornire, come evidenziato da Van Der Kruk et al [1].

Come step successivo, si è pensato di avvalersi di tracciamento GPS, per ricevere informazioni globali e quindi poter derivare caratteristiche delle traiettorie. Ma in questo caso, sorgono altri problemi. Infatti, oltre alla difficile elaborazione dei dati forniti, di cui non si è ancora raggiunto il consenso sulla gestione, vi sono troppe variabili che possono andarne a influenzare l'entità. Alcuni dei quali, affermano Malone et al [2], sono: posizionamento sul corpo o sul mezzo usato dall'atleta, qualità del segnale satellitare e metodi di filtraggio e analisi dei dati.

Primo fra questi risulta essere la qualità del segnale. Essendo la canoa uno sport svolto in esterna e in cui può capitare di essere schermati dall'acqua, non è possibile garantire una chiara comunicazione tra il rilevatore GPS ed i corrispondenti satelliti.

La chiarezza del segnale è poi ulteriormente complicata dall'impossibile controllo sulle condizioni atmosferiche, che variando costantemente, impediscono la garanzia di un ambiente controllato in cui svolgere le analisi.

Successivamente, si è pensato all'utilizzo di realtà aumentata, ma anche in questo caso la presenza dell'acqua e la dinamica dello svolgimento di allenamento risulta essere un problema. Come sostengono Loia et al [3], la realtà aumentata è stata già utilizzata nell'ambito del football americano, permettendo agli atleti di avere informazioni in diretta su posizionamento di compagni e avversari e, di conseguenza, avere informazioni su quali potrebbero essere le migliori strategie da applicare o i migliori movimenti da compiere. In questo caso però, si sta parlando di uno sport di squadra, in cui la tecnica del singolo è relativamente meno importante, ma soprattutto si tratta di uno sport sulla terra.



Figura 2.2: Dispositivo di AR: Hololens 2

La canoa, svolgendosi lungo un canale, causa prima di tutto il rischio che l'atleta possa girarsi e quindi andare sott'acqua, portando così all'immersione anche i dispositivi eventualmente indossati. In più, pensando di far indossare il sistema di realtà aumentata ad un osservatore esterno come un allenatore, si incorrerebbe nel problema dei contrasti e riflessi causati dall'acqua stessa. Non essendo un piano o una superficie con geometria chiara e identificabile, il sistema avrebbe molta difficoltà a capire dove si trova l'atleta rispetto al background, rendendo così un'analisi praticamente impossibile.

La scelta dello strumento da utilizzare è ricaduta quindi sulla computer vision e object detection. Avendo a disposizione le 5 camere installate lungo il percorso, sono disponibili filmati, da diverse inquadrature, di discese di allenamento. Con esse, è stato possibile costruire un database, su cui poi andare ad addestrare un'intelligenza artificiale atta al tracciamento degli atleti. Avendo quindi la disponibilità

di elaborare video, anche in diretta, degli allenamenti, attenzione si è posta a fornire soluzioni sia da un punto di vista registico, in modo da migliorare l'esperienza di pubblico, arbitri e allenatori in egual modo, sia da un punto di vista di analisi della traiettoria. Così facendo, sono state ottenute informazioni impossibili da ottenere altrimenti, che potranno incrementare la qualità degli allenamenti e portare ad un miglioramento delle performance.

Tra le varie reti neurali disponibili, la scelta è ricaduta sulla rete YOLO (You Only Look Once), sviluppata da Ultralytics. Essendo un progetto open source, la rete messa a disposizione è in continua evoluzione ed è strutturata appositamente per computer vision e tracciamento in generale. Per questo lavoro si è utilizzata, nello specifico, la versione 8.

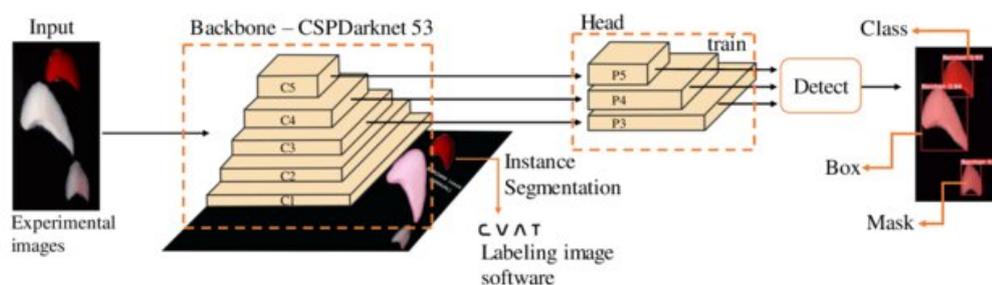


Figura 2.3: Struttura semplificata di YOLOv8 [4]

La versione 8 rispetto alla precedente versione 5, non è soltanto un miglioramento, ma introduce nuove feature e architetture che garantiscono l'elevata accuratezza della rete, testata sui 2 benchmark più popolari, COCO e Roboflow 100. Il primo è un dataset di larga scala contenente oltre 250.000 immagini al cui interno sono presenti 80 classi di oggetti, utilizzato principalmente per valutare la performance di modelli di object detection in scene composte da molti elementi. Roboflow 100, essendo composto da 100 sottoinsiemi di immagini provenienti da differenti ambiti come medicina e sport, valuta la capacità di generalizzazione della rete su diversi task specifici[5].

Sul dataset COCO la v8 ha raggiunto una precisione media del 53.9%, rispetto al 50.7% raggiunto dalla v5[6]. Inoltre, fornisce un pacchetto Python facile da utilizzare e modificare. Citata oltre 20.000 volte in articoli accademici e applicata in diversi scenari reali, rappresenta la miglior rete per le applicazioni necessarie a questo lavoro di tesi.

Capitolo 3

Metodi

Per lo sviluppo di questo lavoro di tesi è stato principalmente utilizzato il linguaggio di programmazione Python, su piattaforma pycharm. I pacchetti costituenti la fondamenta del programma sono stati PyTorch, PyQt5 e ultralytics.

PyTorch rappresenta il framework più utilizzato quando si deve lavorare con machine learning o computer vision. Il fatto di essere open source garantisce un costante aggiornamento da parte di sviluppatori e programmatori anche indipendenti. Pytorch permette di lavorare con strutture di dati chiamati "tensori", il cui significato è solo superficialmente legato a quello matematico. Ciò che questi tensori rappresentano non sono altro che array multidimensionali. Il vantaggio risiede però nella possibilità di farli lavorare in parallelo utilizzando la GPU del pc come componente principale di elaborazione, piuttosto che la CPU, che al contrario è ottimizzata per un'elevata velocità di calcolo, ma sequenziale.

PyQt5 è stato il pacchetto scelto per il design dell'Interfaccia Utente (UI). Esso permette la creazione di layout personalizzati in maniera grafica, semplicemente trascinando i componenti voluti all'interno di una zona rappresentante quella che sarà l'interfaccia finale. Suddetti componenti possono essere: bottoni, finestre contenenti immagini, blocchi di testo, solo per citarne alcuni. Terminato il processo di design, è poi possibile esportare il proprio lavoro sottoforma di codice Python, scritto automaticamente ed ulteriormente modificabile. Infatti sarà necessario inserire manualmente tutte le logiche dell'interfaccia, come le funzioni da svolgere una volta premuto un tasto o la serie di funzioni che permettono di mostrare un video nella finestra prescelta.

Infine, per l'utilizzo delle reti neurali, è stato scelto il pacchetto messo a disposizione da ultralytics. Anch'esso open source, permette di avere accesso alla gestione personalizzata di una grande varietà di versioni di reti neurali, anche di differenti

dimensioni. La specializzazione di queste Intelligenze Artificiali (AI) riguarda l'ambito della computer vision e dell'object detection. E' infatti possibile addestrare le reti con dataset personalizzati sulle proprie necessità, andando a modificare una moltitudine di parametri che permetteranno all'utente di raggiungere il risultato desiderato. Per questo lavoro sono state testate varie dimensioni della rete YOLOv8.

Per l'ottenimento di un dataset adeguato è stata di fondamentale importanza la partecipazione dell'Ivrea Canoa Club. Lungo il canale di discesa sono presenti 5 telecamere inquadranti il percorso, che permettono di avere prospettive uniche durante gli allenamenti degli atleti. Utilizzando le registrazioni di discese fornite, è stato possibile catturare le immagini che sono poi andate a comporre il dataset. In questo modo, è stata anche garantita un'accettabile varietà di inquadrature e differenti condizioni di illuminazione date dalle condizioni atmosferiche.

La stessa rete, addestrata sullo stesso dataset, è stata poi utilizzata per svolgere l'analisi della traiettoria. Per un atleta di canoa è importante raggiungere il traguardo nel minor tempo possibile e spesso, dal lungo canale, osservando a occhio nudo una discesa, è difficile dire in con quale angolo viene approciata una porta. Ma questo approccio, espanso a tutte le porte presenti lungo il percorso, va a condizionare significativamente il tempo finale, in quanto un ingresso sbagliato in una porta precedente potrebbe causare un errore o una penalità in una porta successiva. Per dare supporto agli atleti, si è quindi deciso di identificare graficamente la traiettoria sul canale. Questo è stato possibile utilizzando gli stessi pacchetti citati precedentemente e andando ad elaborare le coordinate del bounding box, accessibili durante il processo di tracking dell'atleta.

3.0.1 Addestramento della rete neurale

La fase preliminare di addestramento della rete è sicuramente l'aspetto più importante del progetto, in quanto da essa ne dipende la qualità dei risultati, nonché la loro accuratezza. Essa inizia dando alla rete il materiale su cui deve essere addestrata e che deve rispecchiare ciò che si vuole analizzare.

Nel caso della presente tesi il materiale utilizzato è stato di tipo fotografico, proveniente da database e non, in cui vengono raffigurate canoe da diverse inquadrature. La scelta dei file di addestramento è ricaduta sulle immagini dovendo successivamente lavorare su feed video in diretta, analizzando singoli frame. Le foto sono state poi marcate andando ad identificare la zona di interesse tramite riquadratura.

Successivamente alla marcatura, le foto sono state date in input alla rete per la fase di addestramento, divise in training set, test set e validation set. In questo modo, la rete è stata addestrata su una parte di immagini diversa da quella utilizzata per verificarne i risultati. Così facendo si è potuta analizzare la capacità di generalizzazione della rete e la qualità dell'identificazione, ovvero se la precisione fosse accettabile o meno.

Durante il processo di addestramento la rete va a modificare i pesi dei collegamenti tra un layer e il successivo, ovvero va a variare il valore per cui ogni neurone viene moltiplicato prima di essere valutato di nuovo nel layer successivo. La distribuzione dei pesi risultante viene restituita sotto forma di file di testo e, dopo molteplici prove, diversificando i parametri di input, è stata selezionata quella che restituiva il miglior compromesso tra generalizzazione e precisione dei risultati.

3.0.2 Database fotografico

Una piccola parte del materiale utilizzato proviene dal database "OPEN IMAGES V7", una raccolta online gratuita messa a disposizione da Google. All'interno di questo database si possono trovare milioni di immagini, divise per categoria di soggetti rappresentati o per tipologia di marcatura effettuata. E quest'ultimo è uno dei vantaggi di questo database. Infatti le immagini che possono essere trovate sono già marcate dagli utenti e identificano un'enorme varietà di soggetti. Le immagini stesse possono essere poi scaricate, selezionando esclusivamente il file necessario ai propri scopi: object detection, segmentazione o svariati altri.

La restante parte del database è stato ottenuto utilizzando la rete di telecamere messe a disposizione dal centro sportivo di canoa slalom di IVREA. Avendo ricevuto una serie di registrazioni, effettuate dal centro stesso, sono stati catturati multipli screenshot, con l'obiettivo di avere quanta più varietà possibile a livello di inquadrature delle canoe. Attenzione particolare è stata posta all'acqua. Essendo essa un elemento di disturbo, è risultato necessario avere varietà sia dal punto di



Figura 3.1: Immagine tipo scaricabile

vista di condizione del percorso di allenamento, sia dal punto di vista di condizioni atmosferiche. Al variare di quest'ultime infatti, le proprietà di riflessione dell'acqua variano, potendo portare a falsi positivi o falsi negativi nell'analisi finale.

3.0.3 Processo di marcatura

Lo step successivo è la marcatura delle immagini. All'interno di esse è necessario andare a comunicare alla rete neurale qual è la parte di interesse che vogliamo che venga identificata. Questo può essere fatto in molteplici modi variabili in accuratezza e differenti a seconda degli scopi del lavoro. La segmentazione ad esempio, è un processo nel quale il soggetto di interesse viene isolato dal resto dell'immagine, andandone a delimitare i confini rispetto al background. Questo processo risulta essere ottimo a livello di capacità di identificazione, ma essendo computazionalmente pesante, non è consigliato per svolgere object detection, a meno che non si disponga di pc molto potenti o costruiti ad hoc.

Il metodo utilizzato in questa tesi è stata la marcatura tramite riquadro, grazie al sito "cvat.ai". Su questa piattaforma è possibile caricare il proprio database di immagini ed effettuare una marcatura personalizzata. Obiettivo della marcatura è identificare al meglio il soggetto di interesse. Inizialmente si è pensato di prendere in considerazione solo l'atleta o solo la canoa. I risultati però non sono stati soddisfacenti, in quanto entrambi, data la posizione delle telecamere e le elevate rotazioni e traslazioni compiute, molto spesso venivano coperti dall'acqua del canale, rendendoli irriconoscibili. Per questo motivo è stato preso in analisi il complesso atleta-canoa. Così facendo l'identificazione è risultata più semplice date le maggiori dimensioni e l'integrazione del contrasto con l'acqua all'interno della zona di analisi.



Figura 3.2: Riquadratura soggetto

Una volta caricata la foto sul sito, si può procedere alla marcatura semplicemente tracciando un riquadro sulla zona di interesse. In background verranno memorizzate le coordinate di due angoli opposti del riquadro in termini di pixel. Tutte le coordinate di tutte le immagini potranno essere poi esportate e scaricate in multipli file di testo, a ognuno dei quali viene assegnato il nome dell'immagine corrispondente. Prima del download risulta possibile decidere il formato di esportazione ed è stato scelto quello compatibile con la rete YOLO.

Nel caso di questo progetto il database completo è composto di 569 immagini, divise in 102 per test e validation set e 365 per training set. All'interno del training set inoltre sono presenti 22 immagini in cui il soggetto di interesse è assente. Questo per sfruttare una feature del processo di addestramento della rete YOLO che permette di dare in input immagini rappresentanti il background, in modo da rendere l'analisi più precisa, senza la necessità di utilizzare quantitativi importanti di foto.

3.0.4 Parametri di addestramento e risultati

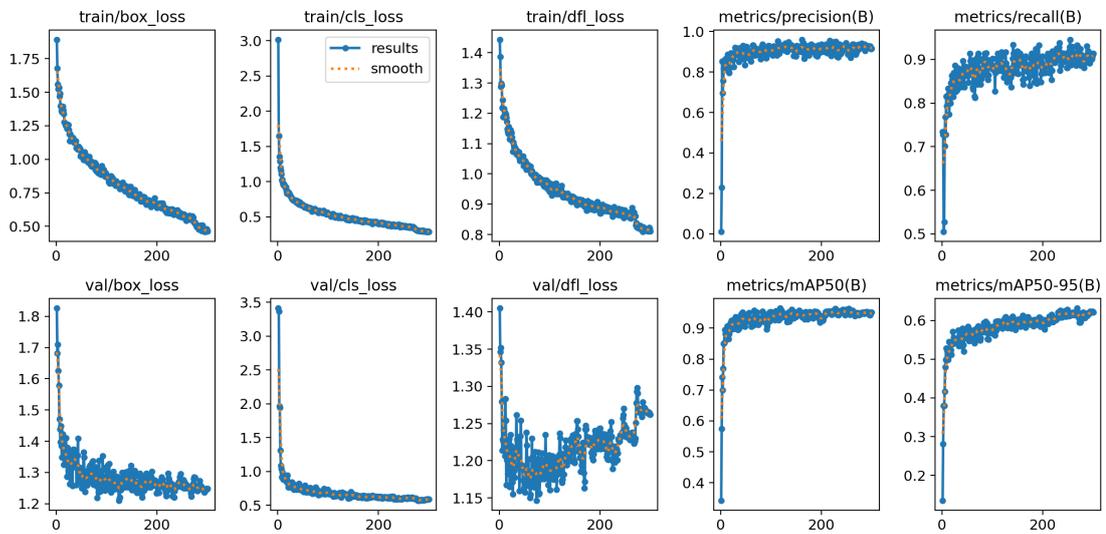


Figura 3.3: Grafici dei risultati di addestramento

Al termine di ogni ciclo di addestramento, vengono restituiti i grafici soprastanti, utili per comprendere se la rete ha raggiunto gli obiettivi voluti. In ascissa abbiamo il numero di epoche, che va impostato facendo attenzione ad evitare overfitting, se troppo alto, e addestramento non sufficiente, se troppo basso. In questo caso il valore scelto è stato di 300, visti i soddisfacenti risultati raggiunti. In ordinata abbiamo la magnitudine dell'errore tra il bounding box reale e quello predetto dalla rete. L'errore totale, chiamato "loss", è dato dalla media pesata dei valori di box loss, cls loss e dfi loss.

Queste ultime caratteristiche sono le stesse presenti nei 6 grafici sulla sinistra e vengono valutate sia sul training set che sul validation set. Esse si definiscono come:

box loss Misura l'errore in termini di coordinate e dimensioni del bounding box reale rispetto a quello predetto. Un teorico valore di 0 significherebbe che i due bounding box sono perfettamente sovrapposti.

cls loss Comunica l'entità dell'errore di classificazione. In questo caso, avendo una sola classe da identificare oltre al background, ovvero l'atleta, si nota come il valore, nella prima decine di epoche, diventi estremamente basso.

dfi loss Rappresenta un parametro di nuova implementazione nelle reti YOLO e indica l'abilità della rete di riconoscere traslazioni o rotazioni della classe

voluta. E' evidente come, data l'elevata mobilità dei soggetti, questo risulti il più difficile da ottimizzare.

Nei 4 grafici presenti sulla destra si va invece a valutare un altro set di parametri. Precision e recall possono essere definiti rispettivamente come la capacità della rete di evitare falsi positivi e come la capacità di assegnare la classe corretta in qualsiasi istanza. La Mean Average Precision (mAP) è invece l'area sottesa alla curva precision-recall. Va quindi ad incapsulare in un solo valore entrambe le performance.

Un'importante misura non mostrata nei grafici, ma utilizzata per valutare l'entità di mAP50 e mAP50-95, è l'Intersection over Union (IoU). Essa quantifica la sovrapposizione che c'è tra il bounding box predetto e quello reale. Di conseguenza mAP50 comunica la capacità di identificazione con una soglia di IoU di 0.50, ovvero su target "facili", mentre mAP50-95, comunica l'identificazione per target più "difficili".

In sintesi, i primi grafici, partendo da sinistra, con parametri definiti da "loss" a denominatore vengono considerati migliori se sono di bassa entità, al contrario degli ultimi quattro che più alti sono, migliore è la performance.

Come si può vedere, nonostante l'utilizzo di un database relativamente piccolo, la tendenza dei valori risulta positiva. Si noti come i primi arrivino quasi tutti allo 0, ma non lo raggiungano: questo perchè raggiungerlo significherebbe che la rete identifica esattamente il soggetto, ma ciò comporterebbe il rischio di overfitting e quindi di una capacità minore di generalizzazione.

Al fine di ottimizzare al meglio l'addestramento è possibile agire su alcuni parametri all'interno del codice, propri della rete neurale. In questo lavoro di tesi, tra le decine di parametri modificabili, si è deciso di agire su: epochs, optimizer, lr0, erasing e close mosaic.

Epochs Il numero totale di epoche di addestramento. Ogni epoca effettua un'analisi dell'intero database. In questo modo si possono gestire sia i tempi di addestramento che la performance del modello. Il valore scelto è stato di 300, per avere una buona via di mezzo tra tempistiche e performance.

Optimizer L'ottimizzatore che si vuole utilizzare. Tra le varie scelte disponibili si è optato per l'ottimizzatore ADAM (Adaptive Moment Estimation). Si struttura come una combinazione di due metodologie di discesa del gradiente: il momento e la propagazione della radice quadratica media. Risulta molto efficiente nel momento in cui si ha a che fare con database molto grandi o con molti parametri. In questo caso, è stato utilizzato in previsione di un futuro ampliamento del database.

Lr0 Il learning rate iniziale, ovvero quanto i pesi vengono modificati durante l'addestramento. Valori elevati velocizzano l'addestramento a discapito della

capacità di riconoscimento, al contrario, valori bassi comportano lunghi tempi di addestramento, portando ad una maggiore precisione. Il valore scelto è stato di 0.001

Erasing Durante il processo di addestramento viene rimossa una porzione random di alcune immagini per forzare la rete a concentrarsi su caratteristiche meno ovvie e variegate. In un range di 0.0 - 0.9 è stato impostato il valore di 0.7 data la piccola porzione di inquadratura occupata dall'atleta.

Close mosaic Per le ultime N epoche effettua data augmentation, ovvero copie delle trasformazioni sul dataset. Le trasformazioni possono essere rotazioni, scalature, traslazioni o cambiamenti di colore. Si è deciso di effettuare questo processo sulle ultime 30 epoche.

Altro elemento fondamentale per il corretto funzionamento della rete è la scelta della sua dimensione. Il team di YOLO mette a disposizione 5 dimensioni per la versione 8: nano, small, medium, large ed extra large.

Ognuna di queste reti ha hidden layer sempre maggiori ed incrementa il numero di neuroni per ogni layer, di conseguenza un aumento di dimensione comporta sì, un incremento nella precisione della previsione, ma allo stesso tempo aumenta i tempi di addestramento e di lavoro, una volta messa in campo.

Per questa applicazione, dopo aver testato reti di dimensioni nano, small e medium, si è optato per l'utilizzo di una rete nano. La scelta è derivata dalla velocità di addestramento elevata e dai risultati soddisfacenti. L'incremento di dimensione non ha portato ad un miglioramento nel tracciamento rilevante. Inoltre, dovendo analizzare 5 flussi video contemporaneamente, quindi avendo 5 reti che lavorano allo stesso tempo, i tempi di analisi dei video sono rimasti abbastanza bassi da permettere un output video a 30 frame al secondo, il minimo richiesto per avere fluidità.

3.1 Regia

Avendo valutato accettabili i risultati di addestramento, è necessario effettuare alcuni test su video per scongiurare eventuali bug o imprevisti che possono scaturire dall'applicazione. In questo capitolo si vedranno i suddetti test e la creazione di un ambiente che rappresenti, almeno in forma rudimentale, quello di applicazione finale. Come materiale di test sono stati scelti file mai visti prima dalla rete, per avere risultati più realistici possibile.

Per lo sviluppo di una UI è stato utilizzato il programma PyQt. Tramite un ambiente di sviluppo grafico è possibile costruire un'interfaccia utente intuitivamente. Successivamente è possibile esportare l'interfaccia in formato di codice, per poi essere implementata all'interno del proprio script. Essendo però ancora soltanto uno scheletro, le connessioni logiche e i trigger di eventuali pulsanti vanno creati ad hoc.



Figura 3.4: Esempio di tracciamento

Mostrato sopra è il semplice output della rete quando le viene dato il compito di analizzare un video. In realtà, ciò che viene analizzato è il singolo frame che poi viene mostrato a schermo. Questa operazione è possibile utilizzando un ciclo while che rimane attivo fintanto che un frame riesce ad essere letto. Anche durante l'analisi è possibile impostare alcuni parametri per ottimizzare al meglio la rilevazione. In questo caso è stata utilizzata l'impostazione "persist" e il valore di "vid stride" è stato impostato a 30.

La prima permette di dire alla rete che ogni frame è temporalmente collegato e che quindi ci si trova davanti ad elementi persistenti. Se così non fosse, la

rete considererebbe ogni frame come un caso isolato, aumentando il tempo di elaborazione. Il parametro "vid stride" permette di effettuare l'analisi a specifici intervalli di frame. Essendo il lavoro svolto su atleti di canoa, la loro velocità risulta relativamente bassa, rendendo possibile analizzare 1 frame ogni 30, ovvero 1 al secondo, senza perdere di precisione. La fluidità del video è poi ottenuta interpolando i risultati di frame successivi, garantendo quindi di avere 30 frame al secondo in uscita.

Dall'immagine si vede come la rete ponga un bounding box intorno al soggetto che pensa essere l'atleta. Il box non ha dimensione fissa, ma, durante il video, si modifica cercando di incapsulare quanto più precisamente possibile l'atleta. Si vede inoltre un numero al di sopra del bounding box. Esso rappresenta la fiducia con cui la rete identifica il bersaglio. Valori che superano lo 0.60 sono accettabili, soprattutto considerando il relativamente piccolo database di addestramento e le ridotte dimensioni della rete utilizzata.

Assicurato il corretto funzionamento della rete nei riguardi di un singolo video, si è passati all'analisi di molteplici video contemporaneamente. Per fare ciò però è risultato necessario sviluppare un'interfaccia utente (UI) comprensibile.

3.1.1 Design UI

Per soddisfare i fini registrati è stato necessario avere la possibilità di visualizzare i 5 flussi video insieme ad un sesto flusso che avrebbe rappresentato ciò che un ipotetico spettatore avrebbe visto.

Sulla sinistra troviamo i feed delle telecamere, nominati con le posizioni sul canale delle telecamere stesse, dopo che sono stati analizzati dalle reti, mentre sulla destra troviamo il video che viene mostrato come risultato dell'analisi. Inoltre, sempre sul lato destro possiamo vedere una serie di tasti che ci permettono di avere un minimo di controllo sulla gestione della diretta.

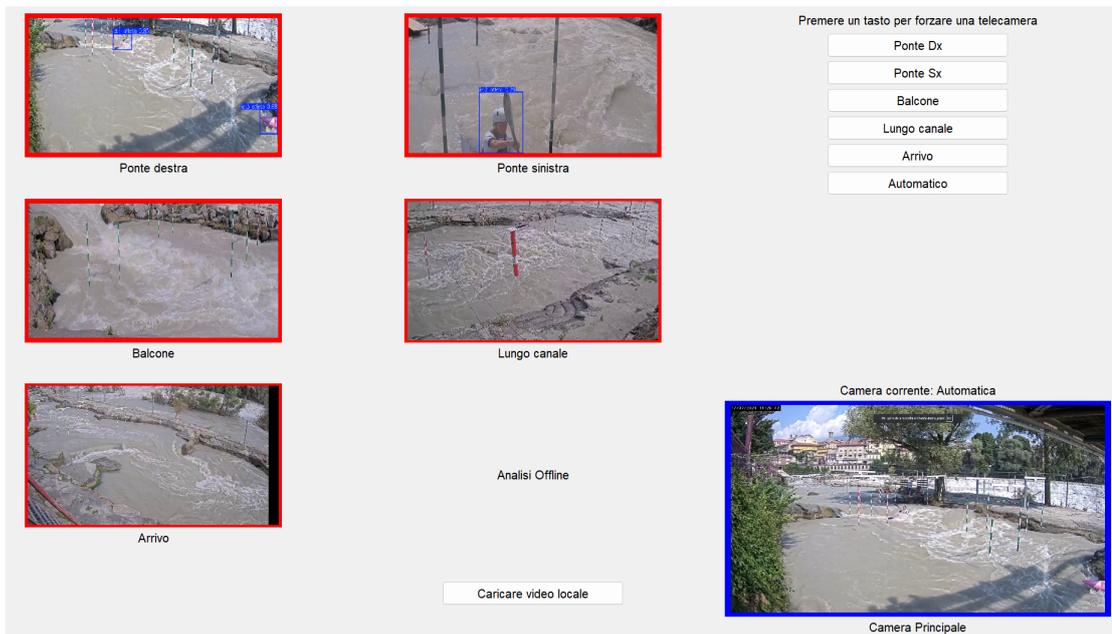


Figura 3.5: Organizzazione UI

La possibilità di gestire la diretta è importante per scongiurare eventuali imprevisti. Il sistema è pensato per un'applicazione in ambito principalmente competitivo, di conseguenza viene fatta l'assunzione che sul percorso di discesa, in qualsiasi momento, ci sia un solo atleta.

Garantendo ciò, la camera principale è in grado di mostrare solo le inquadrature in cui un atleta è effettivamente presente. Ma nel caso ci fosse un incidente, o per qualsiasi motivo si rendesse necessario prendere il controllo, è possibile farlo attraverso i tasti. Essi infatti permettono di sovrascrivere ciò che viene mostrato dall'IA e impostare manualmente la telecamera mostrata. Per poi tornare alla gestione automatica basta premere il tasto "Automatico".

Come ultimo elemento presente nella UI abbiamo un settimo flusso video, questa volta utile se si vuole effettuare un'analisi di una traiettoria, senza uscire dal programma. Premendo il pulsante viene permesso all'utente di selezionare un video da analizzare dai file locali del pc.

3.1.2 Gestione di flussi video multipli

Avendo 5 flussi video contemporanei è risultato necessario far lavorare il programma in parallelo. La struttura generale si compone di una parte principale, che chiameremo "master", e di parti secondarie che chiameremo "slave". Il master si occupa della presentazione della UI e della visualizzazione dei frame, mentre gli slave hanno il compito di svolgere l'elaborazione e l'analisi dei flussi. Il processo può essere sintetizzato nei seguenti punti:

1. Il master crea la UI, la mostra all'operatore e procede con l'avviamento degli slave. In questa parte viene anche gestita la logica dei pulsanti.
2. Ogni singolo slave acquisisce il video, sia esso locale o online in diretta, e attiva l'analisi tramite rete YOLO. Per ogni frame svolge l'identificazione dell'atleta, ne traccia il box e immagazzina il numero di atleti correnti. Al termine di un ciclo di analisi emette verso il master un segnale contenente l'immagine pronta ad essere mostrata ed un secondo segnale che comunica il termine del ciclo.

Vi è la presenza di uno slave ulteriore, che non svolge quanto precedentemente descritto, ma si occupa di segnalare al master quale camera deve essere mostrata come principale, nel caso di selezione tramite tasti.

3. Il master riceve i segnali contenenti le immagini e li elabora in multiple funzioni. 5 funzioni sono adibite all'aggiornamento del feed delle singole camere (che devono essere sempre mostrate), mentre una sesta funzione viene utilizzata per aggiornare il feed principale a seconda della presenza o meno di un atleta.

La priorità è sequenziale, dalla camera 1 all'inizio del percorso alla 5 al traguardo: nel caso sia presente un atleta nella camera 1, essa verrà mostrata come principale, ignorando l'eventuale presenza di atleti lungo il resto del percorso.

Solamente quando l'atleta si sposterà dall'inquadratura 1 alla 2, azzerando quindi le rilevazioni nella prima camera e aumentando di 1 quelle nella seconda, allora la camera 2 verrà mostrata come principale. Questa logica si applica per tutte le camere. Di conseguenza, ad esempio, per avere l'inquadratura 4 come principale, si dovrà avere 1 rilevazione nella camera 4 e nessuna rilevazione nelle camere precedenti.

4. Una volta che tutti i frame dei vari flussi vengono mostrati, il master rimane in attesa di ricevere il segnale di ciclo terminato dagli slave. Ricevuto quest'ultimo, da l'ok per procedere con un altro ciclo.
5. Nel caso in cui il pulsante di analisi di traiettoria venga premuto, viene attivato un settimo slave, che al termine del lavoro tornerà ad essere inattivo.
6. Per catturare la traiettoria percorsa, viene preso il centro del bounding box tracciato dall'IA in termini di pixel e se ne immagazzinano le coordinate in un file di testo successivamente accessibile. Al termine del video, il file di testo viene letto e mostrato come una sequenza di punti, interpolati da linee per creare una traiettoria continua.

Nell'immagine sottostante è mostrata una rappresentazione grafica della struttura del programma.

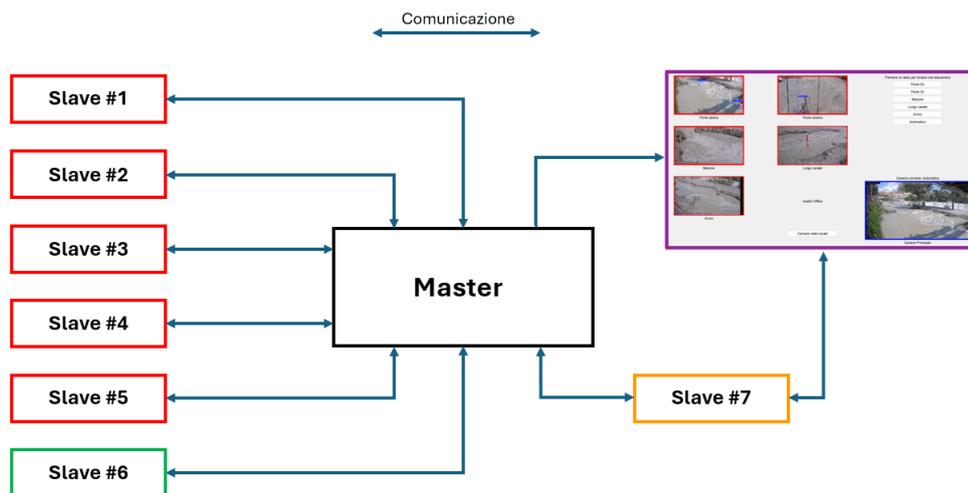


Figura 3.6: Schema strutturale semplificato

A questo punto è chiaro che un grande collo di bottiglia per le performance è l'hardware a disposizione. La continua comunicazione tra master e slave, la gestione della logica della camera principale, unitamente al continuo lavoro di 5 reti neurali, va a pesare sulla potenza di calcolo del pc. Per questo la maggior parte del lavoro viene delegato alla scheda video piuttosto che alla CPU, essendo essa più efficiente nello svolgimento di processi in parallelo.

3.1.3 Logiche interne e risoluzione problematiche

Tutto ciò che è stato descritto nella sezione precedente viene svolto ogni circa 20-40 millisecondi, di conseguenza è fondamentale avere una precisa coordinazione tra le parti comunicanti, per evitare problemi dovuti a conflitti. La maggiore criticità risiede proprio nel garantire che ogni frame parta dallo slave e arrivi ad essere mostrato senza che ci siano interferenze.

Comunicazione master - slave

Per assicurare una continua e fluida comunicazione tra corpo principale e corpo secondario del codice, tutte le interazioni vengono gestite da segnali, categorie di variabili disponibili all'interno del pacchetto PyQt. Utilizzando questo formato, l'informazione, una volta che lascia lo slave, viene dimenticata dalla fonte, permettendo una rapida comunicazione. Da parte del ricevitore, nel nostro caso il master, è importante però che questa informazione non venga modificata durante la sua elaborazione.

Bisogna in altre parole evitare che, mentre l'immagine sta per essere mostrata, non cambi, a causa della ricezione del successivo frame. Per scongiurare questa eventualità, è stato utilizzato un altro set di segnali unicamente adibiti al controllo della prontezza di ogni slave. In questo modo un nuovo frame verrà inviato al master solamente nel momento in cui tutte le immagini saranno state mostrate.

Gestione aggiornamento camera principale

Discorso simile può essere fatto per la camera principale. Qui abbiamo 2 blocchi logici diversi: uno per la gestione automatica e uno per la gestione forzata tramite selezione manuale. Per la gestione automatica, prima dello step di visualizzazione, una funzione controlla una lista, continuamente aggiornata, contenente le rilevazioni di ogni camera. Allo stesso tempo riceve costantemente tutti i frame provenienti da tutti gli slave.

Sulla base quindi delle priorità date alle camere, decide quale delle 5 inquadrature mostrare. Da notare che in questa sezione viene mostrata l'immagine grezza, senza il bounding box posto dalla rete neurale. Questo perchè, in uno scenario realistico, il pubblico non ha necessità di informazioni riguardanti le rilevazioni e così facendo si alleggerisce anche la mole di dati che devono essere trasmessi in ogni istante.

La gestione manuale della regia avviene invece con un sistema di flag, che di default ha valore 0, rappresentante quindi la gestione automatica. Nel momento in cui un utente seleziona una camera premendo un pulsante, uno slave andrà a modificare il valore del flag, associandogli il valore identificativo della camera

selezionata. Il valore verrà quindi letto e verranno mostrati soltanto i frame provenienti dalla camera con identificativo corrispondente.

Lag e falsi positivi

Durante il processo di sviluppo del programma, si sono riscontrati problemi di lag nei video in uscita, dove essi spesso avevano scatti e alcune volte rimanevano bloccati anche per 4-5 secondi. Le cause di questo problema sono state identificate e quasi completamente risolte dove possibile, viste le multiple origini.

La prima fonte di lag risiede nei video originali utilizzati. Sia nel caso di registrazioni che nel caso di video in diretta, ogni 5-6 secondi vengono persi un paio di decine di frame, comportando un salto nel video. Ciò è dovuto sicuramente al mancato accesso ai video in locale. Infatti, a meno di essere connessi al pc utilizzato per accedere alle telecamere, si è sempre lavorato online, di conseguenza la qualità del flusso video è dipendente dalla banda a disposizione. Se quest'ultima non risulta essere sufficiente, o ha dei picchi bassi, la mole di dati necessaria non riesce ad essere trasmessa, portando quindi a degli scatti. Questa fonte di lag è facilmente eliminabile, prelevando i video direttamente dalla sede o aumentando la larghezza di banda a disposizione.

Per quanto riguarda invece fonti interne al codice, ne sono state identificate 3: il tempo impiegato per processare un'immagine e compiere l'identificazione, la comunicazione non efficiente tra master e slave e l'ottimizzazione del codice. Il primo caso è sicuramente il più importante tra i tre.

Bisogna tenere a mente che, per permettere una visualizzazione abbastanza fluida, tutto il processo di classificazione deve rimanere sotto i 30 ms secondi, anche per dare tempo allo slave di inviare il frame al master. Per ottenere questo risultato sono stati applicati molteplici accorgimenti. Primo fra tutti la riduzione di risoluzione di elaborazione.

Essendo i video provenienti dalle telecamere in 1080p, il livello di dettaglio risulta essere troppo elevato e anche non necessario. In primis quindi si è andati a ridurre la risoluzione dei frame a 640p. Si è scelto questo livello perchè, oltre a essere il livello consigliato da ultralytics, garantisce velocità di elaborazione soddisfacenti, avendo meno pixel da analizzare. Inoltre, si è andati a considerare 1 frame ogni 30, lasciando alla rete il compito di interpolare i risultati, che risulta essere un processo più efficiente rispetto all'analisi di ogni singolo frame.

Successivamente, si è identificato un altro collo di bottiglia nella comunicazione tra master e slave, riguardante i tempi di invio/ricezione dei segnali e di applicazione

all'interfaccia dei frame voluti. Dal lato comunicazione, la riduzione di risoluzione ha contribuito alla velocità garantendo un'inferiore mole di dati da inviare. Avendo poi a disposizione uno spazio limitato nell'interfaccia, anche la risoluzione di 640p è risultata eccessiva nel momento in cui in una singola schermata sono presenti 7 video differenti. Si è quindi deciso di applicare un'ulteriore riduzione, portando la risoluzione finale, a 426 x 240, che comunque fornisce un'ottima chiarezza, velocizzando ulteriormente i tempi necessari alla visualizzazione.

Infine, per riuscire ad eliminare qualche millisecondo in più, attenzione è stata posta alla scrittura del codice. L'utilizzo di alcune funzioni, soprattutto quando si parla di Python, comporta un incremento garantito dei tempi di elaborazione. Essendo, in alcuni casi, l'implementazione di cicli while e cicli for obbligatoria, si è provveduto a dividere il più possibile i vari step in funzioni.

Far svolgere molteplici azioni in una singola funzione, risulta infatti essere un processo più lento rispetto ad utilizzare multiple funzioni, a causa del minor tempo di permanenza delle variabili nella memoria volatile del pc. Un vantaggio secondario che si ottiene adottando questo metodo, sta nella possibilità di definire funzioni una singola volta, ma utilizzarle in vari punti del codice, apportando le dovute modifiche.

Durante il periodo di sviluppo, anche il problema dei falsi positivi rilevati dalla rete è stato incontrato e in buona parte risolto. Una prima origine è stata identificata in un problema a monte della rete, riguardante la compatibilità con diversi componenti hardware. Risulta infatti che le reti YOLO hanno un bug per il quale, durante la messa in campo, identificano ogni pixel del video fornito come la classe voluta con il 100% di sicurezza, come mostrato nella figura sottostante. Al momento della scrittura della tesi, questo bug non è ancora stato risolto, ma è stato identificato che avviene solamente su pc che possiedono una scheda video AMD.

In ultimo, la presenza di falsi positivi, tramite iterazioni di addestramento con parametri sempre più restrittivi, è stata in buona parte risolta. Occasionalmente si possono ancora verificare, ma questi sono dovuti alla relativamente bassa quantità di immagini presenti nel database di addestramento e alle limitazioni poste dai componenti hardware a disposizione: l'incremento di epoche o di iterazioni porta ad un incremento troppo significativo dei tempi di addestramento. Ovviamente, avendo a disposizione pc costruiti ad hoc, anche questo problema può essere praticamente eliminato.



Figura 3.7: Bug identificazioni eccessive

3.2 Analisi della traiettoria

Come ultimo lavoro, si è deciso di fornire agli atleti informazioni riguardanti la traiettoria compiuta durante la discesa. Bisogna ricordare che l'occhio umano, soprattutto per oggetti in movimento, è molto limitato a livello di percezione di dettaglio rispetto all'ambiente. L'utilizzo delle stesse telecamere, permette di avere informazioni molto più dettagliate, che possono fornire informazioni utili al miglioramento della performance degli atleti.

I principali modi in cui, durante la discesa, in ottica di gara, è possibile ricevere penalità o perdere secondi preziosi, sono principalmente tre: approccio sbagliato ad una porta rossa, contatto con una porta o salto di porta. Nel primo caso, un errore di ingresso può comportare elevati tempi di percorrenza, anche dipendentemente dalla condizione fisica dell'atleta. Gli ultimi due casi, comportando rispettivamente una penalità di 2 e 50 punti (nella canoa slalom punti = secondi), possono essere mitigati mantenendo una traiettoria corretta e ottimale durante tutte le fasi della discesa.

Per acquisire le prime informazioni si è lavorato in termini di pixel. Le coordinate degli angoli del bounding box sono state estratte durante l'analisi della rete e immagazzinate in un file di testo. Successivamente, considerando i valori medi delle suddette coordinate, si è identificato il centro del bounding box. Per avere un raffronto visivo è stato poi disegnato, sull'ultimo frame, il percorso compiuto

dal centro, andando ad interpolare i punti tramite linee che danno un'ottima approssimazione del tracciato reale. Avendo queste informazioni a disposizione, si è deciso di dare la possibilità di comparare visivamente due traiettorie diverse e si è iniziato il lavoro di conversione in coordinate reali.



Figura 3.8: Visualizzazione traiettoria

Un esempio di applicazione della possibilità di comparare due traiettorie è il seguente: durante l'allenamento l'atleta compie varie discese utilizzando approcci diversi alle porte. Le discese saranno cronometrate e le migliori risulteranno quelle con tempo di percorrenza minore. Andando a fare un confronto tra le discese con tempi inferiori, si avrà la possibilità di identificare i punti in cui si è perso o guadagnato tempo, permettendo di prendere gli approcci migliori singolarmente ed unirli in un'unica discesa.

A questo scopo il programma lavora con la stessa rete usata per la regia. Bisogna semplicemente avere a disposizione una serie di registrazioni delle discese e caricarle nel programma quando richiesto. Durante l'elaborazione sarà possibile vedere la traiettoria che in diretta appare a schermo. Successivamente, ripetendo il processo di caricamento con un video differente, le due traiettorie rimarranno a vista, permettendo una comparazione.



Figura 3.9: Visualizzazione confronto

Step successivo risulta essere la conversione da coordinate intese come pixel a coordinate riferite allo spazio reale. Il problema diventa quindi di svolgere un passaggio da uno spazio 2D ad uno spazio 3D, in cui però la coordinata Z è relativamente influente.

3.2.1 Correzione della distorsione

Innanzitutto è necessario compiere una compensazione della normale distorsione propria di ogni telecamera. L'immagine che noi vediamo non è mai perfettamente piatta, ma possiede un livello di distorsione dipendente dalle caratteristiche costruttive della telecamera stessa, siano esse le lenti o le ottiche adottate. Tutti i passaggi che seguiranno sono stati testati su telecamere differenti rispetto a quelle installate lungo il canale per problemi di accessibilità a queste ultime.



Figura 3.10: Possibili distorsioni

Per arrivare ad un calcolo semplificato dei parametri intrinseci ed estrinseci della telecamera, necessari per comporre le matrici di trasformazione, abbiamo bisogno di un modello lineare. Ciò in generale non è applicabile in quanto le lenti introducono una distorsione che può essere sia radiale che tangenziale.

Abbiamo quindi bisogno di ottenere i parametri di distorsione, in modo da isolarli in una matrice, e successivamente trattare il modello della telecamera come se fosse lineare. Si inizia acquisendo immagini, attraverso la telecamera, di una scacchiera 8x8. Viene normalmente usata una scacchiera in quanto ha dei punti di riferimento facilmente identificabili e ottimo contrasto tra le varie parti. Le foto dovranno avere la scacchiera posta in quante più orientazioni possibili per garantire una buona precisione nell'elaborazione dei coefficienti. Acquisite le foto può essere utilizzato il pacchetto `opencv` di Python per procedere all'elaborazione.

Essendo un pacchetto mirato all'elaborazione di video e utilizzato nella computer vision, contiene al suo interno un set di funzioni preimpostate atte proprio alla correzione di distorsione. Il processo di elaborazione consiste nel lasciare che il programma predica automaticamente i punti di incontro di 4 quadrati alla volta, due bianchi e due neri alternati, e nel verificarne la precisione. Se l'errore, calcolato anch'esso automaticamente, è troppo elevato, allora sarà necessario aggiungere più

inquadrature, mentre se esso è inferiore a 0.1 allora i parametri restituiti potranno essere considerati attendibili.

Matematicamente abbiamo che le coordinate di un punto (x, y) nello spazio reale, saranno rappresentate da coordinate (x_{dist}, y_{dist}) nello spazio dell'immagine. Nel caso di distorsione radiale queste ultime saranno pari a:

$$x_{dist} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (3.1)$$

$$y_{dist} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (3.2)$$

Mentre nel caso di distorsione tangenziale avremo:

$$x_{dist} = x + [2p_1xy + p_2(r^2 + 2x^2)] \quad (3.3)$$

$$y_{dist} = y + [p_1(r^2 + 2y^2) + 2p_2xy] \quad (3.4)$$

I coefficienti $(k_1, k_2, p_1, p_2, k_3)$ sarà ciò che otterremo in output dalla funzione "cv.calibrateCamera" del pacchetto opencv. Per la verifica della corretta compensazione, avremo in uscita anche le stesse immagini fornite in input, con i punti predetti applicati.

3.2.2 Identificazione parametri intrinseci

Avendo ottenuto i coefficienti di distorsione possiamo ora considerare il modello della telecamera come modello lineare, quindi come se stessimo usando non una lente, ma un piccolo forellino, per ottenere l'immagine. Abbiamo quindi 3 sistemi di riferimento: uno reale, uno riferito al foro ed uno riferito al piano dell'immagine.

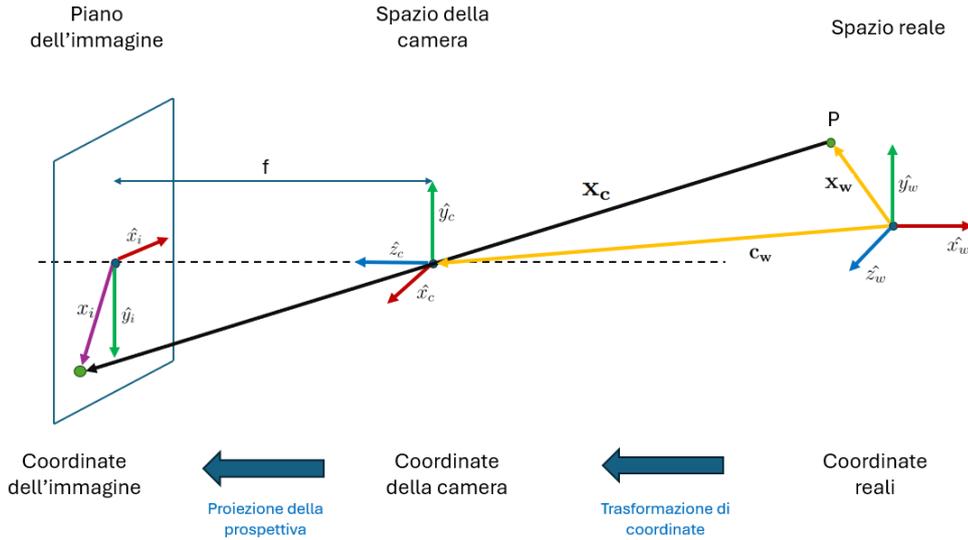


Figura 3.11: Rappresentazione sistemi di riferimento in esame

Nell'immagine soprastante abbiamo anche che f rappresenta la distanza focale effettiva. Conoscendo la posizione corrispondente degli assi del foro rispetto al sistema di riferimento reale, potremo scrivere un'espressione che ci permette di ottenere le coordinate di un punto P nel sistema di riferimento reale, sul piano dell'immagine. Andremo quindi a ragionare al contrario rispetto alle nostre necessità. Una volta conosciuti i parametri, sarà banale invertire il processo e, partendo dalle coordinate sul piano dell'immagine, arrivare alle coordinate reali.

Quello che andremo a fare è prendere un punto $\mathbf{x}_w = [x_w, y_w, z_w]$ nelle coordinate reali e modellarne la sua trasformazione nel sistema di riferimento della camera $\mathbf{x}_c = [x_c, y_c, z_c]$. Ottenuto quest'ultimo applicheremo una proiezione di prospettiva per ottenere le coordinate sul piano dell'immagine $\mathbf{x}_i = [x_i, y_i]$.

Iniziando dalla proiezione di prospettiva sappiamo che:

$$\frac{x_i}{f} = \frac{x_c y_i}{z_c f} = \frac{y_c}{z_c} \quad (3.5)$$

Di conseguenza:

$$x_i = f \frac{x_c}{z_c} y_i = f \frac{y_c}{z_c} \quad (3.6)$$

Che sono le coordinate del punto P proiettate sul piano dell'immagine. Ora, sappiamo che il sensore della camera non ragiona in termini di millimetri, ma in termini di pixel, quindi dobbiamo trovare una corrispondenza tra le due unità di misura. Considerando che, i pixel non sempre sono quadrati, dovremo definire delle densità di pixel m_x e m_y , rispettivamente in direzione x e y , che a noi sono sconosciute, ma sono parte del processo di calibrazione. A questo punto le coordinate di P diventeranno:

$$u = m_x x_i = m_x f \frac{x_c}{z_c} \quad (3.7)$$

$$v = m_y y_i = m_y f \frac{y_c}{z_c} \quad (3.8)$$

Inoltre, ricordando che non conosciamo il punto in cui l'asse ottico penetra il sensore, ovvero l'origine del sistema di riferimento in pixel, non possiamo assumere che esso sia al centro. Solitamente infatti, quando si parla di pixel, si considera l'origine come l'angolo in alto a sinistra per comodità. Allora, considerando il pixel (o_x, o_y) come origine, avremo:

$$u = m_x x_i = m_x f \frac{x_c}{z_c} + o_x = f_x \frac{x_c}{z_c} + o_x \quad (3.9)$$

$$v = m_y y_i = m_y f \frac{y_c}{z_c} + o_y = f_y \frac{y_c}{z_c} + o_y \quad (3.10)$$

Dove i coefficienti m_x, f e m_y, f , essendo entrambi sconosciuti, sono stati combinati in f_x e f_y , considerabili come le distanze focali effettive della camera sull'asse x e y rispettivamente. I parametri (f_x, f_y, o_x, o_y) rappresentano i parametri intrinseci della camera, ovvero ne rappresentano la geometria interna. Per effettuare il passaggio da sistema di riferimento del sensore al sistema di riferimento dell'immagine dovremo trasformare le coordinate (u, v) , in coordinate omogenee, per permettere la trasformazione da 3D a 2D. Trasformare in coordinate omogenee $(\tilde{u}, \tilde{v}, \tilde{w})$ consiste nell'aggiungere un valore fittizio $\tilde{w} \neq 0$ tale che $u = \tilde{u}/\tilde{w}$ e $v = \tilde{v}/\tilde{w}$. Allora, considerato per comodità $\tilde{w} = z_c$, potremo affermare che:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} \equiv \begin{bmatrix} z_c u \\ z_c v \\ z_c \end{bmatrix} = \begin{bmatrix} f_x x_c + z_c o_x \\ f_y y_c + z_c o_y \\ z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (3.11)$$

Otteniamo quindi la matrice contenente tutti i parametri intrinseci della camera, moltiplicata per le coordinate del punto sul piano della camera. La prima è chiamata matrice intrinseca e può essere scritta come:

$$M_{int} = [K|0] = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ con } K = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

La matrice K è chiamata matrice di calibrazione e il fatto che sia triangolare superiore tornerà utile successivamente. Abbiamo quindi ottenuto la matrice M_{int} che ci permette di effettuare la proiezione di prospettiva, ovvero passare dalle coordinate 3D sullo spazio della camera alle coordinate 2D sul piano dell'immagine.

3.2.3 Identificazione parametri estrinseci

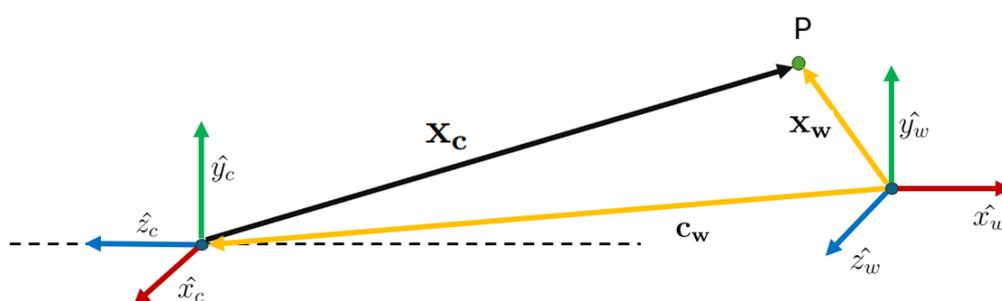


Figura 3.12: Dettaglio spazio camera / spazio reale

A questo punto dobbiamo analizzare il passaggio da coordinate reali a coordinate nello spazio della camera, ovvero dobbiamo effettuare una conversione da 3D a 3D. Questo può essere fatto usando la posizione e orientamento dello spazio della camera rispetto allo spazio reale. La posizione è data dal vettore c_w , mentre l'orientamento è dato dalla matrice R che rappresentano proprio i parametri estrinseci di nostro interesse. La matrice R è una matrice di rotazione in cui ogni riga rappresenta la direzione degli assi x, y e z, rispettivamente, dello spazio della camera rispetto allo spazio reale:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3.13)$$

Avendo quindi i parametri (R, c_w) della camera, potremo trovare il vettore \mathbf{x}_c , rappresentante la posizione del punto P, essendo tutto riferito allo spazio reale, nel seguente modo:

$$\mathbf{x}_c = R(\mathbf{x}_w - \mathbf{c}_w) = R\mathbf{x}_w - R\mathbf{c}_w = R\mathbf{x}_w + \mathbf{t} \quad (3.14)$$

Dove $\mathbf{t} = -R\mathbf{c}_w$ ed è chiamato vettore di traslazione. Scrivendo la 5.4 in forma matriciale, otteniamo:

$$\mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (3.15)$$

Passando come fatto in precedenza in coordinate omogenee:

$$\widetilde{\mathbf{x}}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (3.16)$$

Troviamo la matrice estrinseca M_{ext} :

$$M_{ext} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

A questo punto abbiamo tutto il necessario per passare da coordinate reali espresse in metri a coordinate dell'immagine espresse in pixel. Combinando la M_{int} e la M_{ext} otteniamo la matrice di proiezione completa P :

$$\tilde{\mathbf{u}} = M_{int}M_{ext}\widetilde{\mathbf{x}}_w = P\widetilde{\mathbf{x}}_w \quad (3.18)$$

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \quad (3.19)$$

Step successivo è quindi quello di trovare la matrice P , che poi sarà possibile scombinare per trovare la matrice intrinseca ed estrinseca esplicitamente.

3.2.4 Stima della matrice \mathbf{P}

Consideriamo un oggetto in uno spazio tridimensionale, per comodità un cubo. Possiamo affermare di sapere tutto dell'oggetto in termini di dimensioni e inoltre sappiamo la posizione di ogni punto su di esso. Immaginiamolo avente un pattern a scacchiera per la semplice identificazione dei punti. Poniamo quindi il nostro sistema di riferimento reale in uno degli angoli del cubo. Acquisendo un'immagine, conosciamo quindi le coordinate $\mathbf{x}_w = [x_w, y_w, z_w]$, nel sistema di riferimento reale, e le coordinate $\mathbf{u} = [u, v]$, in termini di pixel. Ovviamente lo stesso ragionamento può essere fatto per qualsiasi punto sul cubo. Otteniamo quindi 2 set di coordinate di punti, uno per i pixel e uno per le coordinate reali. Per ogni punto potremo dire che:

$$\begin{bmatrix} u^{(i)} \\ v^{(i)} \\ 1 \end{bmatrix} \equiv \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x_w^{(i)} \\ y_w^{(i)} \\ z_w^{(i)} \\ 1 \end{bmatrix} \quad (3.20)$$

Espandendo la matrice in equazioni lineari:

$$u^{(i)} = \frac{p_{11}x_w^{(i)} + p_{12}y_w^{(i)} + p_{13}z_w^{(i)} + p_{14}}{p_{31}x_w^{(i)} + p_{32}y_w^{(i)} + p_{33}z_w^{(i)} + p_{34}} \quad (3.21)$$

$$v^{(i)} = \frac{p_{21}x_w^{(i)} + p_{22}y_w^{(i)} + p_{23}z_w^{(i)} + p_{24}}{p_{31}x_w^{(i)} + p_{32}y_w^{(i)} + p_{33}z_w^{(i)} + p_{34}} \quad (3.22)$$

Riscrivendo allora tutte le equazioni lineari e raccogliendole nelle matrici A , di cui conosciamo tutto, e \mathbf{p} , matrice verticale di 12 elementi, a noi sconosciuti, otteniamo:

$$A\mathbf{p} = 0 \quad (3.23)$$

Prima di proseguire, dobbiamo considerare una caratteristica di \mathbf{p} . Prendendo in considerazione un vettore in coordinate omogenee, è noto che, moltiplicandolo per una costante k , il vettore risultante sarà equivalente al vettore di partenza. Il che vuol dire $\mathbf{p} = k\mathbf{p}$, ovvero le due matrici producono esattamente le stesse coordinate omogenee in termini di pixel.

In altre parole, moltiplicare di un fattore k , sia la dimensione della camera, che la dimensione dell'oggetto, implicazione dovuta alla moltiplicazione di \mathbf{p} , non cambia la dimensione risultante dell'immagine. Di conseguenza possiamo scegliere arbitrariamente un valore per k .

Per convenienza scegliamo un valore di k tale per cui $\|\mathbf{p}\|^2 = 1$, che, insieme alla condizione per cui $A\mathbf{p} = 0$, significa trovare il minimo valore possibile di $\|A\mathbf{p}\|^2$ tale per cui $\|\mathbf{p}\|^2 = 1$. Possiamo quindi riscrivere quanto precedentemente detto come:

$$\min_{\mathbf{p}}(\mathbf{p}^T A^T A \mathbf{p}) \text{ tale che } \mathbf{p}^T \mathbf{p} = 1 \quad (3.24)$$

La 5.14 rappresenta un problema conosciuto come "Problema dei minimi quadrati vincolati", di risoluzione nota. Si definisce una funzione di perdita:

$$L(\mathbf{p}, \lambda) = \mathbf{p}^T A^T A \mathbf{p} - \lambda(\mathbf{p}^T \mathbf{p} - 1) \quad (3.25)$$

Che si vuole minimizzare. In altre parole, vogliamo trovare la \mathbf{p} che ci permette di minimizzare L . Per farlo, svolgiamo la derivata di L rispetto a \mathbf{p} ottenendo la semplice equazione:

$$2A^T A \mathbf{p} - 2\lambda \mathbf{p} = 0 \quad (3.26)$$

$$A^T A \mathbf{p} = \lambda \mathbf{p} \quad (3.27)$$

La 5.17 è equivalente alla risoluzione del problema degli autovalori. In altre parole, l'autovettore \mathbf{p} che minimizza la funzione di perdita, non è altro che quello corrispondente al minor autovalore λ della matrice $A^T A$. Trovata l'entità del suddetto autovalore, potremo quindi riarrangiare l'equazione ed ottenere \mathbf{p} .

3.2.5 Decomposizione della matrice \mathbf{p}

Nelle precedenti sezioni abbiamo visto come:

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.28)$$

Dove la prima matrice del prodotto contiene i parametri intrinseci e la seconda contiene gli estrinseci. Considerando le prime 3 colonne della matrice P avremo che:

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = KR \quad (3.29)$$

Ovvero è pari al prodotto tra la matrice K , matrice di calibrazione, ed R , matrice di rotazione, entrambe definite in precedenza. Dato che K è una matrice triangolare superiore ed R è dimostrabilmente ortonormale, è possibile disaccoppiarle a partire dal loro prodotto sfruttando la fattorizzazione QR. Ciò significa che, semplicemente conoscendo la matrice P , possiamo arrivare a conoscere tutti i parametri intrinseci e la matrice di rotazione. Manca però la conoscenza del vettore di traslazione. Osservando però l'ultima colonna di P :

$$\begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = K\mathbf{t} \quad (3.30)$$

In cui conosciamo K e $[p_{14}, p_{24}, p_{34}]$, di conseguenza, invertendo la matrice K :

$$\mathbf{t} = K^{-1} \begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix} \quad (3.31)$$

Arrivati a questo punto, possiamo conoscere tutti i parametri della telecamera e procedere con le conversioni di coordinate, che, una volta graficate, porteranno ad un risultato simile a quello che otterrebbe una vista aerea del canale.

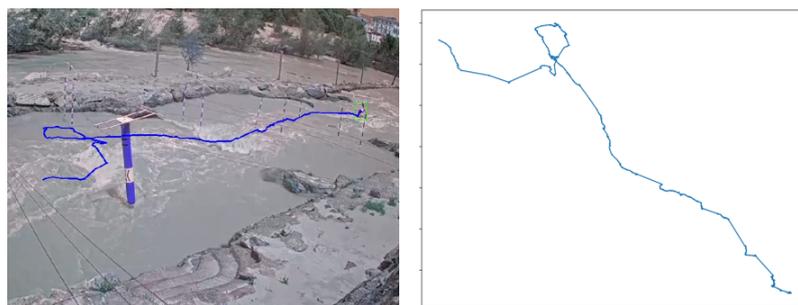


Figura 3.13: Risultato di conversione

Per avere infine un parametro utile al confronto tra due traiettorie, è stato implementato un indice di similarità percentuale. Considerando un valore arbitrario plausibile di varianza per le coordinate x, y , si va a calcolare lo scostamento da una traiettoria presa come riferimento, in riferimento allo scostamento massimo ammesso. In base a soglie preimpostate, il programma andrà a giudicare la nuova traiettoria come molto simile, simile, diversa o molto diversa, valutando ogni punto. Per mancanza di registrazioni su traiettorie paragonabili, la prestazione del codice è stata valutata applicando varianze random ai punti. Si vede come, applicando una varianza inferiore, il sistema comunica che la traiettoria è più simile, rispetto all'applicazione di una varianza maggiore.

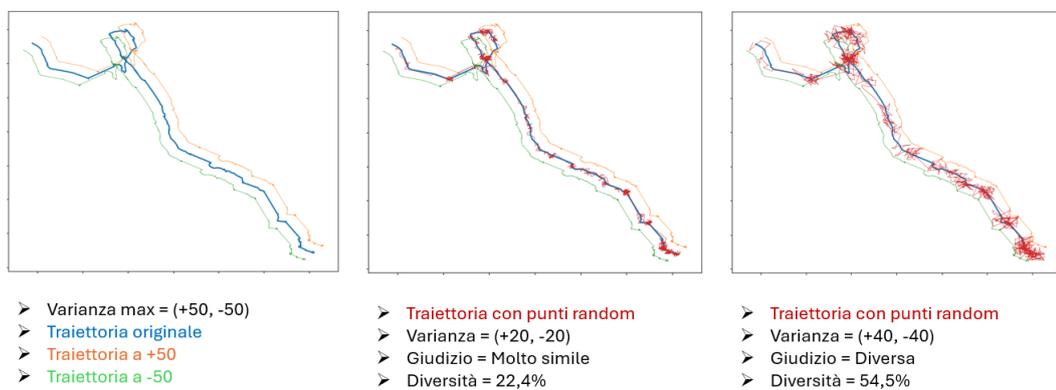


Figura 3.14: Risultati di verifica del corretto funzionamento del programma

Capitolo 4

Risultati

Per quanto riguarda la regia, il sistema riesce a svolgere il suo compito nella maggior parte dei casi. L'uso di multiple reti in parallelo si è rivelato necessario a causa di problemi di comunicazione in caso contrario. Se tutta la mole di lavoro fosse stata assegnata ad una sola rete, essa, non potendo processare più frame contemporaneamente, avrebbe portato ad un delay non trascurabile nella visualizzazione dei video.

Per lo stesso motivo si è deciso di lasciare ad uno slave separato la responsabilità di gestire la logica dei tasti. Inoltre, avere direttamente nell'interfaccia la possibilità di svolgere una rapida analisi di traiettoria, si è rivelato comodo soprattutto per poter formulare giudizi preliminari e riuscire a farsi un'idea sulla qualità della discesa, prima di procedere ad una comparazione più precisa e dettagliata.

Solamente in alcune condizioni particolari, la rete non riesce a seguire perfettamente l'atleta. In primis ciò è causato dal leggero lag ancora persistente nei filmati, a cui però si potrebbe rimediare avendo accesso diretto alle camere o sostituendole con attrezzatura equivalente a frame rate più elevato. Altro problema risiede nell'acqua. Nonostante l'addestramento sia stato soddisfacente nell'ambito del processo di prova del programma, le iterazioni e le ripetizioni vanno sicuramente aumentate.

Capita a volte che la rete pensi che un particolare riflesso sia un atleta, proceda quindi a tracciarlo per qualche frame e poi torni sull'atleta reale. Ciò comporta una perdita di ID assegnato, lampeggiamenti nella camera principale, dovuti al rapido cambio di inquadratura, e la possibilità che venga mostrata un'inquadratura in realtà vuota. Tutte queste complicazioni potranno essere sicuramente ridotte al minimo avendo a disposizione più tempo di addestramento unitamente ad un

database più ampio. Le maggiori iterazioni e la maggiore mole di immagini, porterà ad un affinamento dei pesi della rete, che, di conseguenza, comporterà un significativo incremento nella qualità del tracciamento.

Spostando l'attenzione sull'analisi di traiettoria, si può osservare che il programma, nonostante si trovi in una fase embrionale, riesce a restituire in maniera soddisfacente il percorso effettuato dall'atleta, tanto da poter catturare differenze tra una discesa e l'altra, che a occhio nudo sarebbero praticamente impossibili da identificare.

Ipotizziamo due traiettorie identiche a meno di due minime sezioni, come evidenziato nella figura sottostante. Risaltano immediatamente le zone in cui l'atleta ha effettuato una variazione. In questo esempio, la differenza delle coordinate tra le due traiettorie nelle parti differenti è di soli 5-10 pixel. Ovviamente, nel caso in esame, si sta lavorando con percorsi ipotetici perfettamente identici a causa, come detto nei precedenti capitoli, dell'assenza di materiale adatto. In ottica di applicazione reale, anche una traiettoria uguale all'occhio umano presenterà notevoli differenze quando analizzata e riportata digitalmente. Risulterà quindi molto più semplice, per un allenatore, andare ad associare zone differenti a guadagni o perdite di secondi. E qui risiede uno dei vantaggi di questo sistema.

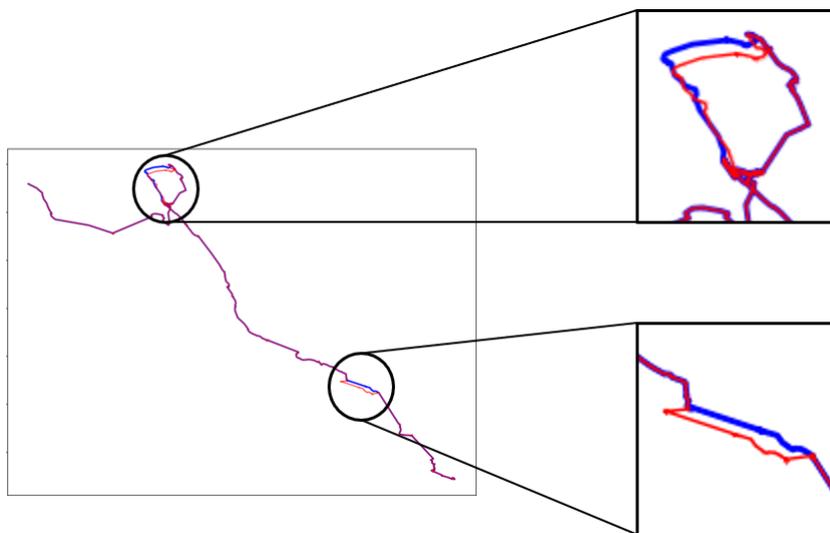


Figura 4.1: Identificazione differenze minime

La possibilità di avere a disposizione questo tipo di informazioni è di fondamentale importanza anche quando si parla di approccio alle rapide presenti lungo il percorso. Durante la discesa, esse rappresentano uno degli ostacoli che, se sottovalutato,

potrebbe rovinare un'intera gara. Vediamo come, l'acqua in movimento nella zona di una rapida, abbia principalmente due direzioni: a favore di corrente e controcorrente. La prima zona è quella centrale, data dal salto che la massa d'acqua deve fare e che ne comporta un'accelerazione verso il traguardo. L'altra si trova ai lati della rapida e spesso a contatto con l'argine del canale.

Questa è una zona di risucchio, in cui l'acqua, espandendosi dal centro verso l'esterno, a causa del precedente salto, la porta a fluire controcorrente per qualche metro. Il fenomeno è dato dal momento posseduto dall'acqua, quando quest'ultima si trova arginata dai bordi del canale: avendo una direzione non perpendicolare all'argine, ma leggermente angolata, data dalla spinta del flusso centrale, tenderà a sfogarsi ed espandersi controcorrente, mostrando un fenomeno paragonabile all'inizio di un mulinello.

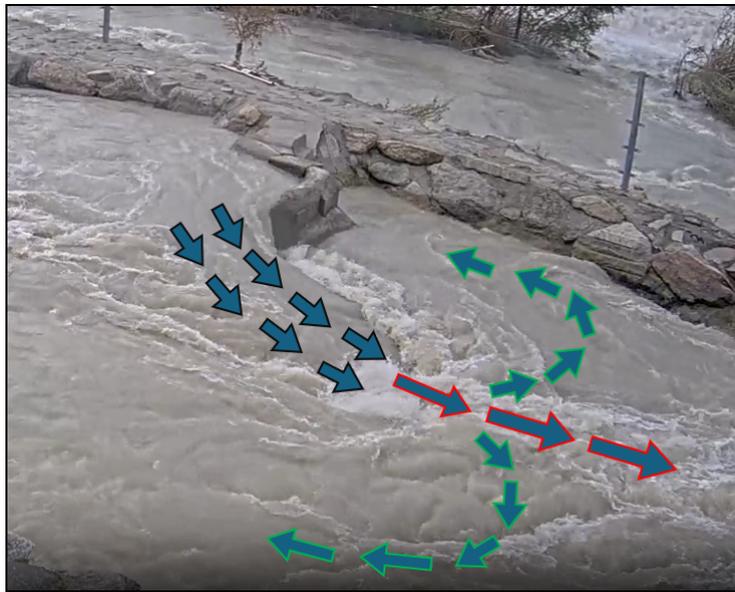


Figura 4.2: Dinamica di una rapida

Risulta importante quindi approcciare le rapide nel modo corretto per evitare che spinte nelle direzioni sbagliate portino ad un maggiore dispendio di energia da parte dell'atleta, con conseguente incremento di tempo di discesa. Si prenda in considerazione l'esempio mostrato in figura 4.3.

Qui abbiamo due approcci leggermente diversi da parte di due atleti durante il Campionato italiano di canoa slalom, svoltosi ad Ivrea il 6 settembre 2020. Le traiettorie rappresentate sono solo qualitative. Superiormente si vede come l'ingresso nella porta è leggermente angolato verso l'interno della rapida. Questo, controintuitivamente, comporta un rallentamento della canoa rispetto ad un approccio diretto, in quanto la canoa si trova in parte fuori dalla massa principale di acqua. Ciò

comporta un passaggio più esterno sulla rapida, che a sua volta trasferisce meno energia, permettendo all'atleta di mantenere una traiettoria stretta. Nell'ultima parte è chiaro come, a causa dei precedenti accorgimenti, la distanza di approccio alla successiva porta rossa è molto ridotta, rappresentando un'ottimizzazione del tempo di discesa.

Spostando l'osservazione alle immagini inferiori, si vede un atleta differente che questa volta approccia la porta verde linearmente, senza quindi dare inizio ad una rotazione della canoa verso l'interno. Questo comporta un maggior tempo di permanenza sulla massa principale d'acqua, che a sua volta causa maggiore velocità guadagnata dall'atleta, anche dimostrata dalla postura che assume per evitare il ribaltamento. L'aumento di velocità non permette il mantenimento di una traiettoria stretta, di conseguenza è chiaro come la traiettoria di approccio alla porta rossa sia peggiore e più difficoltosa, causando maggiore distanza dalla porta e un incremento di energie spese per risalire la corrente.

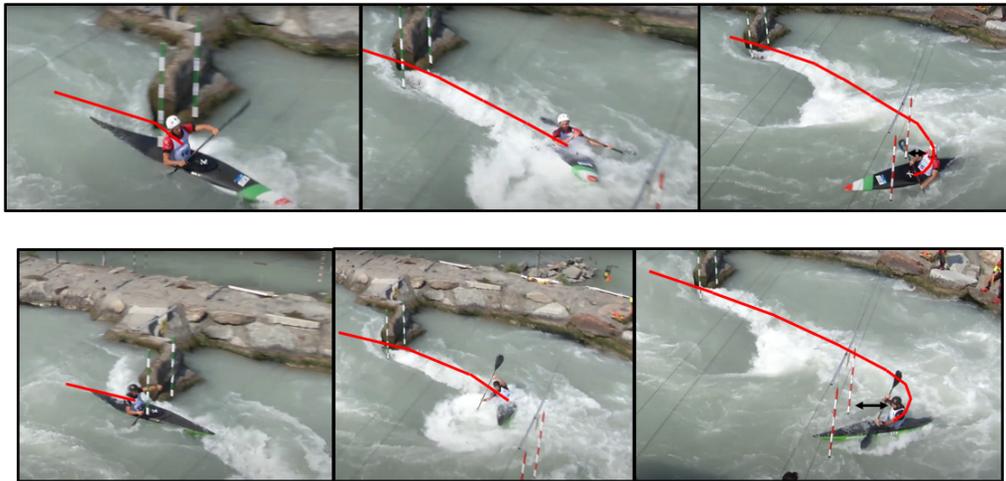


Figura 4.3: Conseguenze di differenti approcci ad una rapida

Per effettuare analisi di questo tipo ci sarà bisogno di elevata precisione nel tracciamento. Sappiamo però che la qualità del tracciamento è subordinata alla qualità dell'addestramento della rete e alla qualità dei filmati forniti in input. Ma l'accuratezza del sistema ha un limite superiore, dovuto alla mole di conversioni che devono essere svolte per passare da coordinate dell'immagine a coordinate di pixel. Durante tutti i passaggi infatti, per forza di cose, vengono svolti arrotondamenti ed approssimazioni che non permettono il raggiungimento di una precisione troppo elevata.

Una soluzione a questo problema potrebbe essere l'utilizzo di telecamere aeree, siano esse fisse o su droni, che osservano il canale sul suo asse z. In questo modo, si

potrebbero trascurare gran parte degli step di conversione, avendo registrazioni già preposte all'analisi, in termini di prospettiva, permettendo quindi il raggiungimento di un'accuratezza ancora più elevata.

Una sessione di allenamento che implementi le tecnologie mostrate in questo lavoro di testi potrebbe avere la seguente struttura: inizialmente sul campo di gara si sistemano le porte in una posizione che cerchi di imitare una situazione di gara, incrementando anche la difficoltà in modo da abituarsi a gestire le peggiori condizioni. Successivamente, l'atleta, andrebbe a svolgere qualche decina di discese senza ancora analizzarle, ma catalogandone i tempi.

Il sistema dedicato alla regia verrà usato per ottenere solo registrazioni utili, dove quindi appare l'atleta in esame, tutte le altre situazioni, come presenza di altre persone o assenza di soggetti, verranno scartate. A questo punto, si procederebbe con l'analisi, graficando tutte le traiettorie svolte una sovrapposta all'altra ed evidenziando la migliore, ovvero quella con tempo di percorrenza minore. Sarà poi compito dell'allenatore e dell'atleta stesso individuare cosa ha comportato la diminuzione di tempo. Alcune cause potrebbero essere: attraversamento più rapido di una porta rossa, dato da un approccio migliore, o maggiore velocità nei tratti tra una porta e l'altra, dati da una traiettoria più lineare.

In più, prendendo rilevamenti temporali su più tratti del percorso, l'analisi potrebbe essere ancora più specifica, andando quindi a individuare tratti che risultano più veloci, nonostante si trovino in traiettorie con tempi totali più lunghi. Così facendo, si potrebbero prendere tutti gli approcci migliori ad ogni sezione del percorso e unirli in un'unica traiettoria ideale. Trovata questa, sarà compito dell'atleta tentare di riprodurla il più fedelmente possibile.

4.1 Future work

Essendo lo scopo di questa tesi quello di gettare fondamenta solide per futuri aggiornamenti, le possibilità di sviluppo sono molto elevate. A cominciare da applicazioni nell'ambito della sorveglianza e sicurezza dell'accesso al centro di allenamento, fino all'ampliamento ad altri sport concettualmente simili, passando per un miglioramento della qualità delle rilevazioni.

4.1.1 Identificazione atleti

Come prima evoluzione si potrebbe implementare un riconoscimento degli atleti. Nel programma infatti, la rete non è ancora capace di assegnare un id univoco ad ogni atleta, ma, nel passaggio da un'inquadratura all'altra, viene assegnato l'id successivo all'ultimo utilizzato. Per garantire un'identificazione univoca, si potrebbe applicare un codice QR sui caschi degli atleti, personalizzato.

Durante il tracciamento, la rete, leggendo il QR, sarebbe capace di assegnare ad esso un id unico, permettendone il tracciamento ed il riconoscimento lungo tutto il percorso. Inoltre, questa soluzione, potrebbe essere sfruttata dal punto di vista della sicurezza. Immaginando la condizione in cui, per avere un codice QR assegnato, fosse necessario essere iscritti al centro, qualsiasi persona non munita di QR o avente codice non registrato nel database del centro, potrebbe essere riconosciuta come estranea.

4.1.2 Tracciamento multiplo e upgrade camere

Anche come conseguenza di una capacità di identificazione, sarebbe possibile gestire il tracciamento di più atleti durante una stessa discesa, eliminando quindi la condizione per cui, per avere inquadrature successive, il percorso precedente debba essere vuoto. Riuscendo ad aggiungere tale funzionalità, il programma potrebbe essere messo in campo anche durante gare di kayak, in cui generalmente 4 atleti effettuano una discesa in contemporanea, permettendo applicazioni registiche anche in questo caso.

Ovviamente, per rimanere al passo con l'aumento di rilevazioni necessarie, si potrebbe aumentare il numero di telecamere installate e utilizzarne alcune di profondità. Incrementandone il numero si avrebbero a disposizione una maggiore quantità di inquadrature permettendo una migliore qualità della rilevazione e la possibilità di analizzare le traiettorie da più punti di vista. L'utilizzo di telecamere di profondità andrebbe a migliorare la raffinatezza delle analisi della traiettoria. Avendo già in se l'informazione della profondità, si potrebbe riportare la traiettoria

in maniera molto più fedele, considerando che, in ogni passaggio di conversione, è inevitabile che si abbia una perdita di precisione, anche se minima.

4.1.3 Utilizzo droni

Evoluzione più avanzata potrebbe anche riguardare l'utilizzo di droni stazionari per ottenere registrazioni maggiormente fedeli. Ponendone un numero adeguato, circa 4-5, lungo il canale, si otterrebbero video già nativamente con una prospettiva utile, senza necessità di svolgere la complessa conversione citata precedentemente. Ad oggi anche droni di livello medio, hanno la capacità di rimanere stazionari, andando a simulare quindi una telecamera aerea.

Inoltre, unendo questa tecnologia ad un riconoscimento degli atleti, sarebbe possibile far seguire i singoli canoisti da un drone personale, in modo da ottenere, in un'unica registrazione, la discesa completa e poterne valutare la traiettoria in modo efficiente e continuativo. Ovviamente, una volta raffinata la coordinazione dei droni, essi potrebbero anche essere utilizzati per fini registici durante le competizioni, permettendo al pubblico di osservare la gara con nuove prospettive, di difficile ottenimento altrimenti.

4.1.4 Lavoro al vogatore

La rete YOLO utilizzata nel progetto ha la caratteristica di non essere relegata esclusivamente all'object detection come possibilità di applicazione. Semplicemente costruendo un database differente, potrebbe essere sfruttata anche nelle sessioni di allenamento in palestra. La rete infatti è capace di riconoscere quando è presente un corpo umano davanti a se e, previa marcatura, applicargli uno scheletro virtuale durante il video, andando a tracciare così i segmenti e le articolazioni principali. Essendo il vogatore l'attrezzo che nel modo migliore simula i movimenti svolti durante una discesa di canoa, un'applicazione di questo tipo risulta ovvia.

Ponendo una telecamera fissa che osserva l'atleta durante il movimento, una volta acquisito un buon set di registrazioni, si potrebbero identificare quali elementi o quali condizioni favoriscono una performance migliore. Alcuni di questi potrebbero essere: l'angolo all'interno del quale la schiena oscilla durante il movimento o gli angoli mantenuti dai gomiti e dalle ginocchia durante le fasi di tirata e spinta, rispettivamente.

4.1.5 Ampliamento ad altri sport

Una volta perfezionata l'applicazione alla canoa, ci si potrebbe espandere ad altri sport concettualmente simili come ad esempio la discesa sci. Confrontando i due sport, nella maniera più semplice possibile, si nota come in entrambi i casi parliamo di atleti che devono attraversare delle porte nel minor tempo possibile. Il vantaggio starebbe nel fatto che, ancora una volta, non sarebbe necessario creare una nuova rete ad hoc, ma basterebbe cambiarne alcuni parametri e addestrarla su un nuovo dataset, comportando un importante risparmio di tempo e risorse.

Sarebbe quindi sufficiente ottenere una buona mole di registrazioni e ripetere il processo di marcatura. In questo caso, ci sarebbe anche il vantaggio di avere a che fare con atleti che si muovono sulla neve, quindi il contrasto tra corpo e background sarebbe facilmente individuabile. Ultimato l'addestramento, la messa in campo sarebbe immediata. La differenza maggiore risiederebbe nel tempo necessario ad ottenere una serie di traiettorie utili per fare un confronto, dati i maggiori tempi di discesa e risalita.

Capitolo 5

Conclusioni

Lo scopo di questo lavoro è stato dimostrare che, grazie ai progressi compiuti in ambito di intelligenza artificiale, è possibile ottenere vantaggi a livello di performance e di qualità della vita molto facilmente. In tutto lo sviluppo della tesi, è stata utilizzata una sola struttura neurale per raggiungere tutti gli scopi esposti. Così facendo, si è eliminato lo step che in passato consumava più tempo, ovvero la costruzione di una rete da zero.

Dal punto di vista della regia, l'applicazione di questo sistema, permetterebbe una minor mole di lavoro da parte degli addetti e una pari, se non migliore, qualità di visione delle competizioni da parte degli spettatori. Teoricamente, non c'è limite al numero di telecamere installabili, l'unica limitazione risiede nell'hardware del pc utilizzato. In ottica di arbitraggio, la possibilità di avere replay istantanei e diversi punti di vista dello stesso attraversamento di una porta, porterebbe la correttezza e la giustizia a nuovi livelli.

Scongiorando errori umani, a volte inevitabili, e potendo sciogliere dubbi o contestazioni, si avrebbero competizioni dove la parità di trattamento degli atleti è garantita. In ultimo, implementando il sistema anche come mezzo di sorveglianza, si avrebbe un accesso al centro più sicuro, soprattutto nel caso in cui si stia svolgendo una competizione di grande importanza, come i Mondiali di settembre 2024, momento nel quale si ha un ingente afflusso di pubblico.

Per quanto concerne gli atleti, il supporto e l'ottenimento di nuove informazioni durante l'allenamento, porterebbe senza dubbio ad un miglioramento delle performance, rendendo l'ambiente ancor più competitivo. La possibilità di studiare in dettaglio le discese effettuate, oltre che a rendere le correzioni da applicare più mirate, ne ridurrebbe i tempi di identificazione. Ovvero, si potrebbe restringere la quantità di variabili che portano ad una performance non soddisfacente. Lo stesso discorso sarebbe ovviamente valido per qualsiasi sport compatibile a cui venisse

applicato questo sistema, ribadendo il fatto che non sarebbe necessario creare una nuova rete, ma fornire a quest'ultima soltanto un database differente.

Punto di forza di questo sistema, risiede nel suo posizionamento all'interno dell'ambiente di allenamento. Adottandone l'utilizzo infatti, gli allenatori rimarrebbero comunque un elemento fondamentale per l'atleta e non sarebbero sostituiti da un corrispettivo tecnologico. Il programma è un mezzo che fornisce maggiori informazioni senza esprimere un giudizio sulla loro qualità, ma presenta un confronto tra di esse.

In altre parole, non dice se una performance è accettabile o meno, ma mette in mano agli allenatori dati che altrimenti sarebbero inaccessibili e che fino ad oggi non sono stati presi in considerazione, lasciando in mano alla componente umana la responsabilità di effettuare valutazioni sfruttando le conoscenze acquisite negli anni. Utilizzandoli al meglio, unitamente all'esperienza degli atleti e dei preparatori, lo sport della canoa slalom potrebbe raggiungere un livello di competizione e di qualità di tecnica mai vista prima.

Elenco delle figure

2.1	Esempi di IMUs	4
2.2	Dispositivo di AR: Hololens 2	5
2.3	Struttura semplificata di YOLOv8 [4]	6
3.1	Immagine tipo scaricabile	10
3.2	Riquadratura soggetto	11
3.3	Grafici dei risultati di addestramento	13
3.4	Esempio di tracciamento	16
3.5	Organizzazione UI	18
3.6	Schema strutturale semplificato	20
3.7	Bug identificazioni eccessive	24
3.8	Visualizzazione traiettoria	25
3.9	Visualizzazione confronto	26
3.10	Possibili distorsioni	27
3.11	Rappresentazione sistemi di riferimento in esame	29
3.12	Dettaglio spazio camera / spazio reale	31
3.13	Risultato di conversione	35
3.14	Risultati di verifica del corretto funzionamento del programma	36
4.1	Identificazione differenze minime	38
4.2	Dinamica di una rapida	39
4.3	Conseguenze di differenti approcci ad una rapida	40

Bibliografia

- [1] Eline van der Kruk e Marco M. Reijne. «Accuracy of human motion capture systems for sport applications; state-of-the-art review». In: *European Journal of Sport Science* 18.6 (2018), pp. 806–819 (cit. a p. 4).
- [2] James J. Malone, Ric Lovell, Matthew C. Varley e Aaron J. Coutts. «Unpacking the Black Box: Applications and Considerations for Using GPS Devices in Sport». In: *International Journal of Sports Physiology and Performance* 12.s2 (2017), S2-18 - S2-26. DOI: 10.1123/ijssp.2016-0236. URL: <https://journals.humankinetics.com/view/journals/ijssp/12/s2/article-pS2-18.xml> (cit. a p. 4).
- [3] Vincenzo Loia e Francesco Orciuoli. «ICTs for exercise and sport science: focus on augmented reality». In: *Journal of Physical Education and Sport* 19 (2019), pp. 1740–1747 (cit. a p. 5).
- [4] Esteban Cúñez Benalcázar e Erick Franklin. «Detection and tracking of barchan dunes using Artificial Intelligence». In: (mag. 2024). DOI: 10.21203/rs.3.rs-3553762/v1 (cit. a p. 6).
- [5] <https://medium.com/@georgekenjiputra/yolo-v8-is-it-better-than-v7-4f978ceaec52> (cit. a p. 6).
- [6] Juan Terven, Diana-Margarita Córdova-Esparza e Julio-Alejandro Romero-González. «A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS». In: *Machine Learning and Knowledge Extraction* 5.4 (2023), pp. 1680–1716. ISSN: 2504-4990. DOI: 10.3390/make5040083. URL: <https://www.mdpi.com/2504-4990/5/4/83> (cit. a p. 6).
- [7] <https://magazine.impactscool.com/robotica-e-ai/>.
- [8] <https://inside.fifa.com/innovation/world-cup-2022/>.
- [9] <https://www.geeksforgeeks.org/>.
- [10] <https://makerfairerome.eu/en/game-set-tech-the-technology-of-tennis/>.

BIBLIOGRAFIA

- [11] <https://www.youtube.com/@firstprinciplesofcomputerv3258>.
- [12] <https://www.federcanoa.it>.