

Facial Exercises Detector: progetto Image and Video Analysis

Salvatore Cerullo
7016502

salvatore.cerullo@stud.unifi.it

Abstract

Al giorno d'oggi è sempre più comune utilizzare degli strumenti di supporto informatici ed intelligenti in molteplici campi applicativi. Uno dei principali campi di applicazione di questi strumenti è quello medico. In questo breve articolo, si vuole presentare una tecnica di analisi video assistita da computer che, sfruttando diverse tecniche tra cui il Metric Learning e il Dynamic Time Warping, è in grado di supportare la riabilitazione di pazienti che hanno disturbi motori che compromettono i movimenti del volto.

1. Introduzione

Oggigiorno nel campo della medicina è sempre più comune ritrovare applicazioni intelligenti in grado di supportare il medico nella diagnosi e nella riabilitazione dei pazienti. Molte di queste applicazioni sfruttano le tecniche di analisi di immagini e/o video.

In questo breve articolo, si vuole presentare una tecnica di analisi video assistita da computer in grado di supportare la riabilitazione di pazienti che hanno disturbi motori che compromettono i movimenti del volto.

In particolare, l'applicazione realizzata sarà in grado, attraverso l'utilizzo di una videocamera, di *osservare* l'esecuzione di un esercizio di riabilitazione facciale e *definire* se è stato svolto in maniera corretta o meno attraverso l'utilizzo di diverse tecniche tra cui il *Dynamic Time Warping* e il *Metric Learning*.

1.1. Riabilitazione facciale

Prima di osservare nel dettaglio l'applicazione software realizzata, verranno introdotti brevemente quelli che sono gli **esercizi di riabilitazione facciale** gestiti da tale applicazione, oltre che i *metodi* e le *tecniche* utilizzate per **valutare** il corretto svolgimento o il fallimento.

1.1.1 Esercizi di riabilitazione facciale

Gli esercizi che verranno trattati sono volti alla riabilitazione di pazienti che, a causa di alcune patologie, come ad esempio la paralisi del nervo facciale, hanno capacità limitate nei movimenti dei muscoli del volto.

In genere questi esercizi consistono nel **portarsi** da una *posizione di riposo* ad una *posizione finale* (fig. 1). E' di fondamentale importanza quindi osservare l'intero movimento che viene eseguito dal paziente e non osservare solo la posizione finale raggiunta.

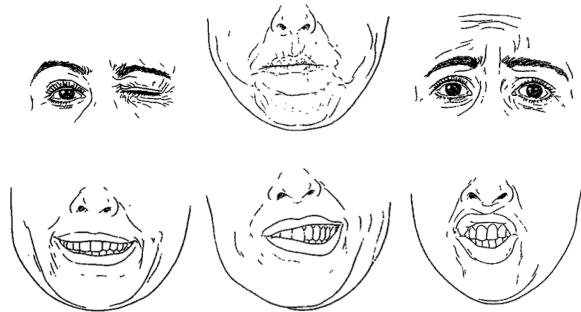


Figure 1: Alcune posizioni finali assunte durante gli esercizi di riabilitazione facciale.

1.1.2 Tecniche di valutazione

Il problema fondamentale dello svolgimento dell'esercizio è la **valutazione**. Infatti, nella maggior parte dei casi, vengono svolti in maniera autonoma dal paziente che deve sempre eseguirli davanti ad uno specchio, in modo tale che sia in grado di valutarne la corretta riuscita o il fallimento.

Nel caso di svolgimento della fisioterapia in maniera autonoma sorge una complicazione: *non sempre un paziente è in grado di valutare l'esecuzione dell'esercizio*. Per tale motivo, nella maggior parte dei casi, questi hanno bisogno di personale qualificato che sia in grado di seguirli durante la riabilitazione. Questo però presenta due prob-

lemi: il primo riguarda un fattore prettamente economico, infatti il personale che dovrà seguire una persona durante un percorso riabilitativo ha un dispendio economico che, in alcune situazioni, può anche essere rilevante; il secondo riguarda le tecniche con le quali si vanno a valutare i movimenti del volto, che nella maggior parte dei casi sono tecniche soggettive.

Le tecniche per valutare una singola esecuzione di un esercizio possono appartenere ad una delle due categorie seguenti:

- **Soggettiva:** la valutazione soggettiva viene fatta sulla base di ciò che un osservatore è in grado di percepire analizzando i movimenti del viso del paziente mentre esegue l'esercizio. A questa categoria appartengono anche le osservazioni effettuate in maniera autonoma dallo stesso paziente. Ovviamente, queste tipologie di valutazioni tendono ad essere distorte e potrebbero non fornire una misura sufficientemente sensibile per il processo decisionale in un determinato contesto clinico;
- **Oggettiva:** le tecniche oggettive cercano di eliminare quanto più possibile gli errori della prima categoria. Esistono diverse soluzioni che sfruttano dei sistemi tecnologici come scanner 3D, visione stereo, ecc. Un'altra soluzione oggettiva è quella che sfrutta una fotografia del volto del paziente in posizione di riposo ed un righello per misurarne gli spostamenti. Attraverso l'utilizzo di queste tecniche si è in grado di misurare con maggiore precisione gli spostamenti, di conseguenza l'analisi dell'esecuzione dell'esercizio è più accurata.

Pertanto, le tecniche oggettive per valutare il movimento facciale sono indispensabili per il trattamento di pazienti con disturbi motori come paralisi facciali, labbro leporino, ecc.

Nel seguito di questo articolo verrà presentato ed analizzato un sistema di analisi del movimento basato sull'analisi di video acquisiti da una singola telecamera in grado di valutare la corretta esecuzione dell'esercizio o il suo fallimento.

2. Approccio al Problema

L'applicazione in esame è un'applicazione che vuole abbracciare e supportare due categorie di utenti:

- **Paziente:** lo scopo principale del sistema è quello di aiutare un utente ad eseguire gli esercizi di riabilitazione facciale in piena autonomia, senza la supervisione di un fisioterapista;
- **Fisioterapista:** oltre alla funzionalità principale vista nel punto precedente, permette ad un esperto di correg-

gere i risultati aggiornando il *dataset di addestramento* e migliorare quindi la precisione dei risultati.

Per poter adempiere alle caratteristiche sopra elencate il sistema svolge diversi passaggi che possono essere suddivise in due fasi principali:

- **Addestramento della metrica:** è una fase preliminare, fondamentale per poter *individuare una metrica di distanza* adatta al problema in esame. Per tale scopo sfrutterà le differenze di posizioni delle feature di due frame, che chiameremo *frame di riposo* e *frame finale*;
- **Esecuzione esercizio:** dopo aver appreso una metrica adatta si procede con la seconda fase. Dapprima è necessario avviare la telecamera e individuare il volto dell'utente frame by frame; a questo punto, per ogni frame avviene l'estrazione dei landmarks facciali relativi al volto individuato; per ogni frame si normalizzano questi landmark in modo da avere invarianza rispetto ad alcune trasformazioni; infine, viene applicata la tecnica del DTW sfruttando la metrica appresa durante la fase di addestramento.

2.1. Riconoscimento facciale

Prima di entrare nel dettaglio dell'implementazione delle due fasi brevemente presentate sopra, si vogliono introdurre i due meccanismi di **riconoscimento facciale** ed **estrazione dei landmark facciali**, che vengono ampiamente utilizzati nelle fasi introdotte precedentemente.

Esistono numerosi metodi di acquisizione del volto in grado di identificare la presenza di uno o più volti in un contesto qualsiasi. La maggior parte di questi è in grado di rilevare pose frontali o quasi-frontali.

Nel caso in esame, per estrarre i volti dai frame, è stata utilizzata la libreria open-source **Dlib** che implementa numerose tecniche di image processing e machine learning. Questa infatti, dà la possibilità di ottenere una *bounding box* per ogni volto orientato frontalmente presente all'interno di un'immagine.

2.1.1 Estrazione dei landmark facciali

Dopo aver rilevato il volto, il passo successivo è l'**estrazione dei landmark facciali**. Questi non sono altro che dei *punti che permettono di localizzare alcune regioni salienti del viso* come ad esempio gli occhi, il naso e la bocca.

Esistono molti tipi di rilevatori di punti di landmark facciali, quello utilizzato per il progetto in esame è quello messo a disposizione nella libreria dlib addestrato sul set di dati **iBug 300-W a 68 punti** e viene utilizzato per stimare le posizioni di 68 coordinate bidimensionali (x, y) che mappano la struttura facciale all'interno di un frame (fig. 2).

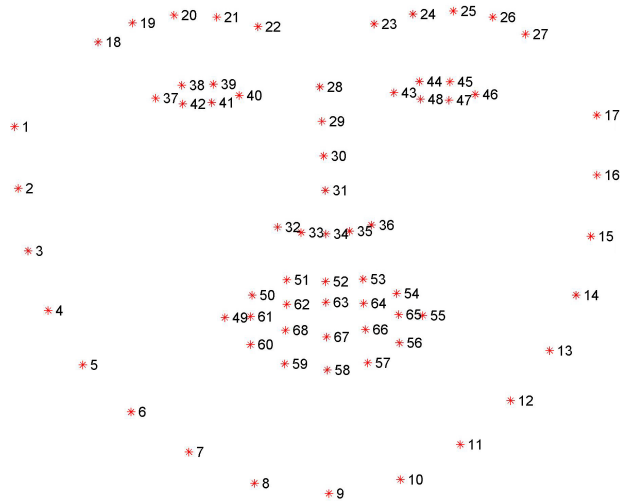


Figure 2: Rappresentazione dei 6 landmarks facciali dal set di dati iBUG 300-W

2.1.2 Normalizzazione

Un'operazione di fondamentale importanza da implementare al fine di avere dei risultati ottimali è la **normalizzazione dei dati**. Nel caso in esame per normalizzazione si intende il processo che ci permetterà di rendere *invarianti* i landmarks rispetto alle trasformazioni di *rotazione*, *traslazione* e *scala*.

Un'altra tipologia di *normalizzazione dei dati* implementata permette di rendere questi dati invarianti rispetto alla morfologia del viso, rendendo quindi l'applicazione indipendente dall'aspetto morfologico dei vari utenti.

L'applicazione quindi prevede due tipologie di normalizzazione che verranno analizzate nel dettaglio di seguito.

ALLINEAMENTO DEL VOLTO

La prima normalizzazione che verrà esaminata è quella che riguarda l'**allineamento del volto**. Come detto in precedenza, questa consiste nel rendere le posizioni dei landmark invarianti rispetto a *rotazione*, *scala* e *traslazione*. In sostanza l'algoritmo acquisirà come **dati di input** le posizioni in coordinate bidimensionali (x, y) dei landmarks e fornirà come **dati di output** ancora una volta dei landmarks ma che rappresenteranno una faccia con le seguenti caratteristiche:

- *Centrata nell'immagine;*
- *Ruotata in modo tale che la linea degli occhi sia perfettamente orizzontale;*
- *Scalata in modo che la distanza del soggetto dalla telecamera risulti irrilevante.*

A tale scopo, l'algoritmo implementato svolge i seguenti passaggi per ogni singolo frame:

1. *Individua la bounding box della faccia ed estrapola le coordinate dei landmarks non normalizzati;*
2. *Individua i centroidi degli occhi;*
3. *Individua l'angolo di rotazione rispetto all'asse orizzontale ed applica la **rotazione**;*
4. ***Scala** le coordinate in base ad una costante di riferimento (nel caso in esame il centro dell'occhio sinistro);*
5. *Si **traslano** tutti i landmarks rispetto ad una costante che rappresenterà il centro della nostra faccia.*

Al termine di questi semplici passaggi avremo dei nuovi landmarks con coordinate (x, y) *normalizzate rispetto alle trasformazioni di traslazione, rotazione e scala*.

NORMALIZZAZIONE MORFOLOGICA

Un aspetto di fondamentale importanza è quello di rendere il sistema **invariante alla morfologia del viso di qualsiasi soggetto utilizzi l'applicativo**. Per morfologia del viso si intende la forma del viso, che come ben sappiamo cambia di persona in persona.

Un primo approccio utilizzato, infatti, prevedeva il confronto frame by frame delle coordinate normalizzate dei landmarks. Questo però dava dei risultati contrastanti se a svolgere l'esercizio era una persona morfologicamente diversa da quelle presenti nei dati di addestramento.

L'approccio che infine è stato utilizzato, e che risolve il problema sopra presentato, non sfrutta le coordinate normalizzate in maniera diretta, bensì sfrutta le **differenze di quest'ultime calcolate rispetto ad un frame di riposo**.

Questa soluzione, poiché tiene conto degli **spostamenti a cui i vari landmarks sono sottoposti durante l'esecuzione dell'esercizio** e non dalle loro coordinate, rende il sistema indipendente dalla morfologia del viso.

2.2. Addestramento

Come detto precedentemente esiste una fase fondamentale che riguarda l'**apprendimento di una metrica di distanza**. Queste operazioni devono essere ripetute per ogni esercizio messo a disposizione dall'applicazione in modo tale che ogni esercizio abbia una metrica definita.

Il primo step per poter addestrare un modello di apprendimento metrico è quello di **popolare il dataset** di un determinato esercizio. A tale scopo l'applicazione segue i seguenti passaggi in ogni esecuzione dell'esercizio:

1. Individuare un frame di riposo estraendone i landmarks normalizzati;

2. Individuare un frame di posa finale estraendone i landmarks normalizzati;
3. Calcolare la differenza di posizione dei landmark di questi due frame;
4. Inserire questi elementi all'interno del dataset;

Quando il dataset dell'esercizio in questione contiene un buon numero di esecuzioni (cioè è popolato), si passa ad applicare la tecnica del Metric Learning con la quale viene appresa la metrica di distanza utilizzata successivamente dall'algoritmo di Dynamic Time Warping.

Andiamo a dare un'occhiata rapida alla tecnica del Metric Learning.

2.2.1 Metric learning

L'apprendimento delle metriche di distanza, detto anche **metric learning**, mira a costruire automaticamente *metriche di distanza* specifiche per attività a partire da dati supervisionati, attraverso dei meccanismi di *apprendimento automatico*. La metrica della distanza appresa può essere quindi utilizzata per eseguire varie attività (es. Classificazione k-NN, clustering, information retrieval ed altro).

Metric Learning supervisionato Gli algoritmi di apprendimento metrico supervisionato prendono come input alcuni punti di input x ed alcune *etichette target* y e apprendono una matrice di distanza che avvicina i punti della stessa classe (per la classificazione) o con valori target vicini (per la regressione) e punti di classi diverse o distanti con valori target molto distanti tra loro.

Dati di input Al fine di addestrare un modello, sono necessari due oggetti *arraylike* (simili agli array), X e y . X potrebbe essere un array 2D di forma $(n_samples; n_features)$, dove $n_samples$ è il numero di punti del nostro dataset e $n_features$ è il numero di attributi che descrivono ogni punto. Ad esempio y potrebbe essere un array 1D di forma $(n_samples;)$ contenente per ogni punto in X la classe a cui appartiene.

Nel caso in esame si ha una situazione analoga all'esempio appena trattato. In particolare, i dati di input sono un insieme lungo $n_samples$ contenente gli "spostamenti dei landmarks tra frame di riposo e frame finale", quindi della forma $(68; 2)$. Il *labels*, invece, è un array lungo anch'esso $n_samples$ che contiene le classi a cui appartengono le varie esecuzioni: se corrisponde ad un'esecuzione sbagliata -1 ; altrimenti $+1$.

2.3. Esecuzione Esercizio

Dopo aver terminato le procedure "preliminari" appena osservate, che consistono nel definire le metriche dei vari

esercizi addestrando i modelli di metric Learning, il sistema è pronto per *analizzare un video di input* e **valutare** se l'esercizio è svolto in maniera *corretta* o *sbagliata*.

Attraverso l'interfaccia, che vedremo nel dettaglio nella prossima sezione, l'utente ha la possibilità di selezionare un esercizio da svolgere. L'applicazione, quindi, avvia la videocamera e chiede all'utente di mantenere una determinata posizione di riposo, durante la quale andrà ad acquisire il frame con la posa iniziale.

Dopo aver individuato il frame di posa iniziale chiederà all'utente di svolgere l'esercizio selezionato ed acquisirà le variazioni di posizione dei landmark presenti nei primi 40 frame validi (frame frontali dai quali dlib riesce ad estrapolare i landmark).

Una volta ottenuti questi dati, che rappresentano l'esecuzione dell'esercizio, andrà a confrontarli con i dati di addestramento all'interno del dataset dell'esercizio in questione calcolandone la distanza. In pratica andrà a misurare la distanza tra l'esercizio svolto e gli elementi presenti nei due set di dati (esempi corretti ed esempi errati) attraverso la tecnica del DTW e sfruttando la metrica da noi appresa per quest'esercizio.

Il risultato dell'esercizio verrà calcolato individuando la minima distanza tra quelle calcolate nel punto precedente e verrà fornito in una finestra di dialogo in cui verrà evidenziata la valutazione dell'esercizio (Svolto in maniera Corretto o Sbagliata), le distanze minime con i due video più simili alla nostra esecuzione ed appartenenti ai due set di dati.

Di seguito andiamo a dare uno sguardo alla tecnica del Dynamic Time Warping.

2.3.1 Dynamic Time Warping

Il **Dynamic time warping**, o *DTW*, è un algoritmo che permette l'allineamento tra due sequenze, e che può portare ad una *misura di distanza tra le due sequenze allineate*. Tale algoritmo è particolarmente utile per trattare sequenze in cui singole componenti hanno caratteristiche che variano nel tempo, e per le quali la semplice espansione o compressione lineare delle due sequenze non porta risultati soddisfacenti. È stato utilizzato in diversi campi di applicazione, dal Riconoscimento vocale, al riconoscimento di attività motorie.

Applicazione del DTW al contesto in esame La soluzione adottata è quella di analizzare tutti i frame acquisiti dalla telecamera in un determinato periodo di tempo, acquisire gli spostamenti dei landmark facciali e successivamente applicare la tecnica del DTW per individuare la distanza tra due esecuzioni.

Una problematica fondamentale riguarda la rappresen-

tazione delle varie esecuzioni sotto forma di dati di input al fine di poterle confrontare attraverso il DTW.

La soluzione adottata è la seguente: ogni video avrà una lunghezza di 40 frame validi (cioè frame contenenti landmarks facciali). Per cui una singola esecuzione viene rappresentata all'interno del sistema da un elemento arraylike avente forma (40, 68, 2), dove 40 è il numero dei frame, 68 è il numero dei landmarks e 2 sono gli spostamenti dei landmarks di un determinato frame rispetto al frame di riposo. La tecnica del Dynamic Time Warping verrà quindi utilizzata per individuare la distanza tra due elementi aventi la forma appena descritta sopra ed utilizzando la metrica di distanza appresa in precedenza.

3. Applicazione

In questa sezione dell'articolo verranno approfonditi due aspetti dell'applicazione realizzata: in un primo momento verrà presentata brevemente la GUI; successivamente vedremo come effettuare alcune operazioni e quindi come utilizzare l'applicazione per eseguire le fasi di *addestramento* ed *esecuzione esercizio*.

3.1. Graphical User Interface

L'intera applicazione è stata programmata utilizzando il linguaggio Python. Per quanto riguarda l'interfaccia grafica è stata sfruttata la libreria **PyQt5**.

Quando l'applicazione viene lanciata si presenta all'utente con una *Main Page* molto semplice (fig. 3) composta da 6 elementi fondamentali:

1. **Selettore di categoria:** attraverso questo selettore a tendina viene scelta la categoria degli esercizi presenti nel dataset;
2. **Selettore esercizio:** attraverso questo selettore è possibile selezionare l'esercizio che si vuole eseguire ed appartenente alla categoria sopra selezionata;
3. **Impostazioni Esercizio:** permette di accedere ad un menu contenente diverse funzionalità per la *modifica* e la *gestione* dell'esercizio selezionato attraverso 1 e 2;
4. **Descrizione esercizio:** breve descrizione dell'esercizio selezionato;
5. **Video esemplificativo:** breve video dimostrativo dell'esercizio selezionato;
6. **Esegui Esercizio:** permette di avviare l'esecuzione dell'esercizio.

3.1.1 Impostazioni Esercizio

Attraverso il pulsante 3 della Main Page è possibile accedere al **Menu impostazioni esercizio** (fig. 4). Andiamo a vedere brevemente le funzioni presentate all'utente:

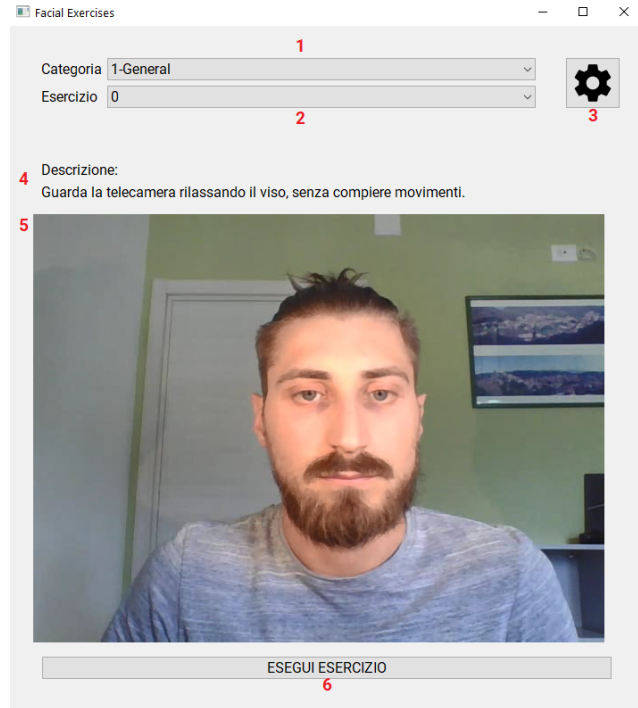


Figure 3: Main Page dell'applicazione.

1. **Nuovo Video:** questi due pulsanti permettono di realizzare un nuovo video da inserire nel dataset etichettato rispettivamente come *CORRETTO* o *ERRATO*;
2. **Rimuovi Video:** mostra all'utente una finestra con la quale è possibile selezionare un video dal dataset e rimuoverlo definitivamente (fig. 5);
3. **Addestra:** premendo questo pulsante verrà richiamata la funzione di apprendimento della metrica di distanza per l'esercizio in questione, sfruttando i dati inseriti nel dataset;
4. **Modifica Descrizione:** permette all'utente di modificare la descrizione dell'esercizio in questione.

3.1.2 Esegui Esercizio

Attraverso il pulsante 6 della schermata principale è possibile svolgere l'esercizio di riabilitazione facciale selezionato. Per tale scopo l'applicazione mostrerà una nuova finestra contenente 4 elementi fondamentali (fig. 6):

1. *Video invariante rispetto traslazione, rotazione e scala;*
2. *Video originale con face detection e landmark facciali;*



Figure 4: Menu impostazioni esercizio.

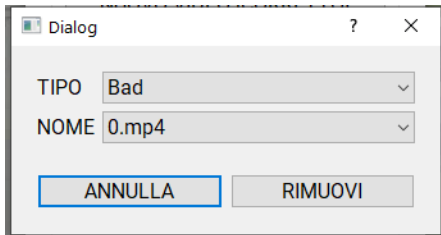


Figure 5: Pagina rimozione video.

3. *Landmark facciali invarianti a traslazione, rotazione e scala;*
4. *Progressbar che indica lo stato di esecuzione dell'esercizio.*



Figure 6: Interfaccia di esecuzione esercizio.

Risultati Al termine dell'esecuzione di un esercizio verrà visualizzata una nuova finestra che presenterà all'utente alcune informazioni sull'esito dell'esercizio (fig. 7):

1. Esito dell'esercizio;
2. Match più vicino con i video presenti nel dataset di video sbagliati e distanza calcolata;
3. Match più vicino con i video presenti nel dataset di video corretti e distanza calcolata;
4. Pulsante che permetterà di effettuare un'operazione di correzione del risultato.

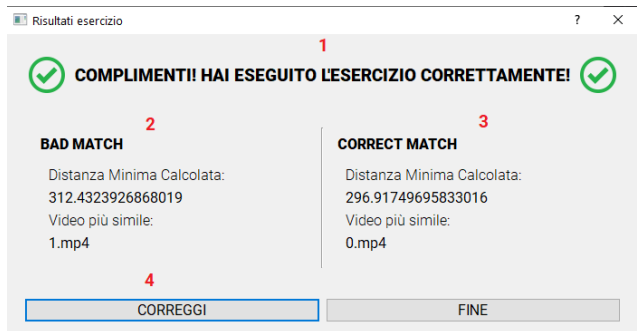


Figure 7: Risultato dell'esercizio.

Correzione Attraverso il pulsante *Correggi* si accede ad una nuova finestra in cui ritroviamo 4 elementi fondamentali (fig. 8):

1. Il frame che indica la posizione di riposo assunta dall'utente all'inizio dell'esercizio;
2. Il frame che indica la posizione finale assunta dall'utente al termine dell'esercizio;
3. Pulsante che permette il salvataggio del video come *Video di comparazione*;
4. Pulsante che permette il salvataggio dei due frame come *Video di training*.

3.2. Utilizzo dell'applicazione

In questa sezione andremo ad analizzare come eseguire alcune operazioni fondamentali gestite dall'applicazione in esame.

Per prima cosa andremo a spiegare brevemente come sfruttare l'applicazione al fine di apprendere una metrica di distanza per un determinato esercizio.

Successivamente osserveremo come eseguire un esercizio e come valutare la sua esecuzione.

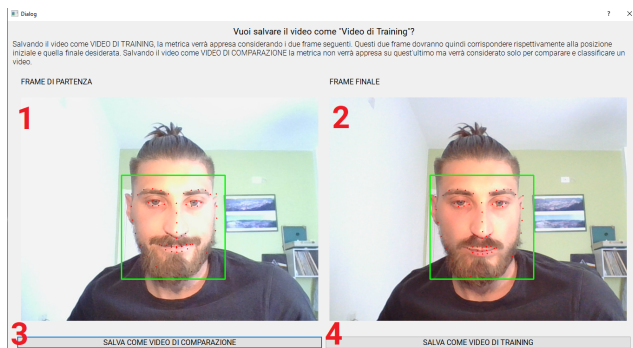


Figure 8: Interfaccia di correzione esercizio.

3.2.1 Apprendimento della metrica di distanza

Supponiamo di aver avviato l'applicazione per la prima volta, e supponiamo che le directory contenenti i dataset degli esercizi siano vuote.

Per rendere l'applicazione utilizzabile dal paziente, bisognerà innanzitutto ripopolare il dataset e successivamente apprendere la metrica di distanza che verrà utilizzata al termine dell'esecuzione di un esercizio.

Ripopolare il dataset Al fine di ripopolare il dataset quello che si deve fare è, innanzitutto, selezionare all'interno della Main Page l'esercizio per il quale vogliamo inserire dei nuovi video di addestramento e successivamente premere il pulsante impostazioni esercizio. Aperto il Menu Impostazioni Esercizio quello che dovremmo fare è realizzare un nuovo video di addestramento. Per fare questo ci basterà cliccare su uno dei due pulsanti in base alla tipologia di video che vogliamo inserire nel database:

- **Nuovo video [Corretto]:** permette la realizzazione di un video in cui un utente svolge l'esercizio in questione in maniera corretta;
- **Nuovo video [Errato]:** permette la realizzazione di un video in cui un utente svolge l'esercizio in questione in maniera errata;

Al termine dell'acquisizione del video, l'applicazione chiede all'utente la **modalità di salvataggio** del video appena realizzato. L'utente ha quindi due opzioni:

- **Salva come video di training:** in questo caso, oltre a salvare il video all'interno della cartella corrispondente all'esercizio in questione, andrà a salvare i frame visualizzati nella schermata come *frame di riposo* e *frame finale*. Solo i video salvati con questa modalità verranno successivamente utilizzati per apprendere la metrica;

- **Salva come video di comparazione:** in questo caso il video verrà utilizzato solo durante la fase di valutazione dell'esercizio. In pratica verrà utilizzato dal DTW per individuare se l'esercizio svolto dall'utente è corretto o sbagliato.

Addestramento Quando si ritiene che il dataset abbia gli elementi sufficienti per apprendere una metrica quello che l'utente dovrà fare è semplicemente ritornare nel menu impostazioni esercizio e premere il pulsante Addestra.

Attraverso questo pulsante infatti, il programma andrà ad analizzare tutti i video salvati come *Video di training* e calcolerà, attraverso la tecnica del Metric Learning, la metrica adatta all'esercizio selezionato.

E' di fondamentale importanza eseguire questa operazione ogni qual volta vengano inseriti nuovi elementi come Video di Training.

3.2.2 Esecuzione dell'esercizio

Dopo aver appreso la metrica di distanza per un determinato esercizio può essere eseguito.

Per fare questo basterà selezionare l'esercizio che si vuole svolgere e premere il pulsante *Esegui Esercizio* nella *Main Page* dell'applicazione.

A questo punto l'applicazione avvierà una nuova finestra ed inizierà ad acquisire le immagini dalla nostra videocamera. Lo svolgimento di un esercizio prevede due fasi: inizialmente verrà chiesto all'utente di mantenere una posizione di riposo (in modo tale da poter acquisire il frame di riposo) e successivamente verrà richiesto di svolgere l'esercizio mantenendo la posa finale fino al termine dell'esecuzione (in modo tale da poter acquisire il frame di posa finale).

Quando l'esecuzione dell'esercizio è terminata, il software svolgerà le operazioni di comparazione tra tutti i video presenti nel dataset sfruttando la tecnica del DTW attraverso la metrica appresa precedentemente.

Comparati tutti i video, visualizzerà la finestra dei risultati, già presentata nella sezione dedicata alla GUI.

Correzione Esercizio Questa sezione del software, così come la parte riguardante le impostazioni dell'esercizio, riguarda delle funzionalità che sono rivolte ad un utente esperto, come un medico o un fisioterapista che è in grado di valutare soggettivamente l'esecuzione di un'esercizio.

Infatti, in questa pagina, l'utente può scegliere di **correggere il risultato** poiché ritiene che la valutazione dell'esercizio è stata svolta in maniera errata dall'applicazione.

Effettuando questa operazione quindi verrà richiesto all'utente se salvare il video corretto come *Video di comparazione* e/o *Video di training*.

Ricordo che se si tratta di una correzione di tipo *training* bisognerà *riaddestrare* il modello della metrica di distanza dell'esercizio in questione.

4. Conclusioni

Per concludere posso affermare che, dopo aver adottato diverse soluzioni per raggiungere gli obiettivi dell'applicazione, l'utilizzo simultaneo delle tecniche di Metric Learning e Dynamic Time Warping ha migliorato molto l'affidabilità del sistema, fornendo una precisione molto elevata rispetto ad altri approcci.

Infatti, in un primo momento in cui era stato utilizzata solo la tecnica di Metric Learning, i risultati erano molto meno precisi ed inoltre era necessario creare dei dataset molto grandi al fine di avere risultati accettabili.

Dal punto di vista degli sviluppi futuri ho individuato diverse funzionalità che potrebbero essere introdotte in modo da rendere il sistema più completo.

Una su tutte potrebbe riguardare la modalità di acquisizione dei frame di posa iniziale e di posa finale che potrebbe essere fatta in modo automatico dal sistema.

Un'altro miglioramento riguarda invece la possibilità di svolgere l'esercizio in un tempo variabile e non fissato a *40 frame validi*.