UNIVERSITÀ
DI PAVIA

FACOLTÀ DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E
DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN BIOINGEGNERIA

TESI DI LAUREA

Eye Tracking per la valutazione dell'inseguimento visivo nella riabilitazione robotica dell'eminattenzione

Candidato: Francesca Pedaci

Relatore: Prof. Stefano Ramat
Correlatore: Ing. Roberto Soldi

A.A. 2025/2026

Indice

Introduzione

L'eminattenzione spaziale, nota anche come *spatial neglect*, è un disturbo dell'attenzione caratterizzato dalla difficoltà a orientare l'attenzione ed esplorare lo spazio controlesionale rispetto alla lesione cerebrale. Nella maggior parte dei casi è associata a lesioni dell'emisfero destro conseguenti a ictus e si manifesta con una ridotta consapevolezza del lato sinistro dello spazio. Poiché può coinvolgere funzioni sensoriali, motorie e cognitive, questa sindrome compromette significativamente l'autonomia del paziente nelle attività della vita quotidiana, rendendo necessario un intervento riabilitativo mirato.

La riabilitazione visiva ha l'obiettivo di favorire il riorientamento dell'attenzione verso lo spazio controlesionale e di migliorare le capacità di esplorazione dell'ambiente. Sebbene non esista attualmente un trattamento universalmente riconosciuto come *gold standard*, negli ultimi anni l'introduzione di tecnologie innovative quali la Realtà Virtuale, l'*Eye Tracking* e i sistemi di *biofeedback* ha contribuito a potenziare l'efficacia delle strategie riabilitative tradizionali. Tra queste, l'*Eye Tracking* consente di monitorare in modo oggettivo i movimenti oculari, fornendo misure quantitative utili alla valutazione del comportamento visivo e dei progressi terapeutici. Parallelamente, anche la robotica ha assunto un ruolo sempre più rilevante nell'ambito riabilitativo. I sistemi robotici possono infatti essere impiegati sia come dispositivi di assistenza fisica sia come strumenti in grado di guidare e supportare il paziente durante il trattamento.

In questo contesto si inserisce la presente attività di ricerca, sviluppata nell'ambito del progetto *Fit for Medical Robotics (Fit4MedRob)*, un Partenariato Esteso finanziato dal Ministero dell'Università e della Ricerca (MUR) nell'ambito del Piano Nazionale di Ripresa e Resilienza (PNRR). Il lavoro è stato sviluppato nell'ambito della *Mission 2 – Activity 5 (Match Making #53),* dedicata allo sviluppo e alla validazione di soluzioni robotiche per la riabilitazione dell'eminattenzione spaziale post-ictus.

L’obiettivo specifico del progetto è utilizzare la tecnologia di *Eye Tracking* Tobii Pro Glasses 3 per monitorare i movimenti oculari durante l'esecuzione degli esercizi di inseguimento supportati dal robot collaborativo TIAGo. Quest’ultimo, nell'ambito di un’iniziativa di matchmaking con la Fondazione Mondino del progetto Fit4MedRob, è stato sviluppato dal laboratorio di Bioingegneria come ausilio per la riabilitazione di pazienti affetti da eminattenzione. In particolare, il presente lavoro si concentra sulla stima dell’ampiezza dei movimenti oculari e dell’errore tra il punto di sguardo del soggetto e il target movimentato dal robot TIAGo lungo traiettorie prestabilite, con l’obiettivo di ottenere misure quantitative utili alla valutazione dell’esplorazione visiva del paziente. Le attività sperimentali sono state svolte presso *il CoE REDI (Centre of Excellence on Rehabilitation Devices and Digital Instruments)* dell'Università di Pavia.

La tesi è organizzata in cinque capitoli. Nel Capitolo 1 viene presentato lo stato dell’arte relativo all’eminattenzione spaziale, con particolare attenzione alle strategie di riabilitazione visiva e all’impiego delle tecnologie robotiche a supporto del percorso terapeutico. Il Capitolo 2 descrive gli strumenti utilizzati per le acquisizioni, in particolare il sistema di *Eye Tracking* Tobii Pro Glasses 3 e il software Tobii Pro Lab. Nel Capitolo 3 viene illustrata l’elaborazione dell’acquisizione di calibrazione e vengono presentati i risultati ottenuti, costituiti dalle rette di calibrazione utilizzate per il calcolo degli angoli oculari. Il Capitolo 4 descrive l’elaborazione delle acquisizioni di inseguimento del target e presenta i risultati relativi al calcolo dell’errore dello sguardo in termini di angoli oculari e all'analisi delle metriche ottenute. Infine, il Capitolo 5 riporta le conclusioni del lavoro e i possibili sviluppi futuri.

Workflow sperimentale

Per lo scopo di questa tesi sono state effettuate due diverse tipologie di acquisizioni: un'acquisizione di calibrazione dello strumento e diverse acquisizioni di inseguimento del target. L'acquisizione di calibrazione richiede al soggetto di osservare in sequenza una serie di target in posizioni angolari note all'interno di una matrice di calibrazione posta a distanza predefinita. I dati ottenuti permettono di determinare la relazione tra le coordinate del punto dello sguardo misurate in pixel dall'*Eye tracker* e la corrispondente posizione angolare, rispetto al soggetto, dei punti osservati, consentendo così di ricavare le rette di calibrazione utilizzate successivamente per il calcolo degli angoli oculari. Le acquisizioni di inseguimento del target, invece, prevedono che il soggetto segua con lo sguardo un target in movimento, guidato da un robot lungo una traiettoria definita a priori. Nel corso del lavoro queste acquisizioni sono state utilizzate con finalità differenti. Una prima parte è stata impiegata per la costruzione dei dataset e per l'addestramento della rete YOLOv8n dedicata al rilevamento automatico del target nel video della telecamera di scena. Successivamente, ulteriori acquisizioni sono state utilizzate per valutare le prestazioni dei modelli addestrati e selezionare quello migliore. Infine, le acquisizioni finali sono state elaborate per ottenere la posizione del target e dello sguardo (*gaze*) frame per frame e, applicando le rette di calibrazione ottenute nella fase precedente, calcolare gli angoli di rotazione dei bulbi oculari durante l'inseguimento visivo. Affinché le rette di calibrazione possano essere applicate correttamente ai dati di inseguimento, è necessario che l'acquisizione di calibrazione e quella di inseguimento vengano eseguite consecutivamente, senza rimuovere gli occhiali di *Eye tracking*. In questo modo si mantengono inalterate le condizioni di acquisizione e la validità della calibrazione del sistema. Nella Figura 1 sono riportate le fasi di acquisizione ed elaborazione relative alle due tipologie presentate. Come si può osservare, la fase di acquisizione è comune a entrambe, mentre differiscono le successive procedure di elaborazione dei dati. Per questo motivo, nelle sezioni seguenti verrà inizialmente descritta la

fase di acquisizione comune, mentre le successive fasi di elaborazione saranno trattate separatamente per ciascuna tipologia di acquisizione.

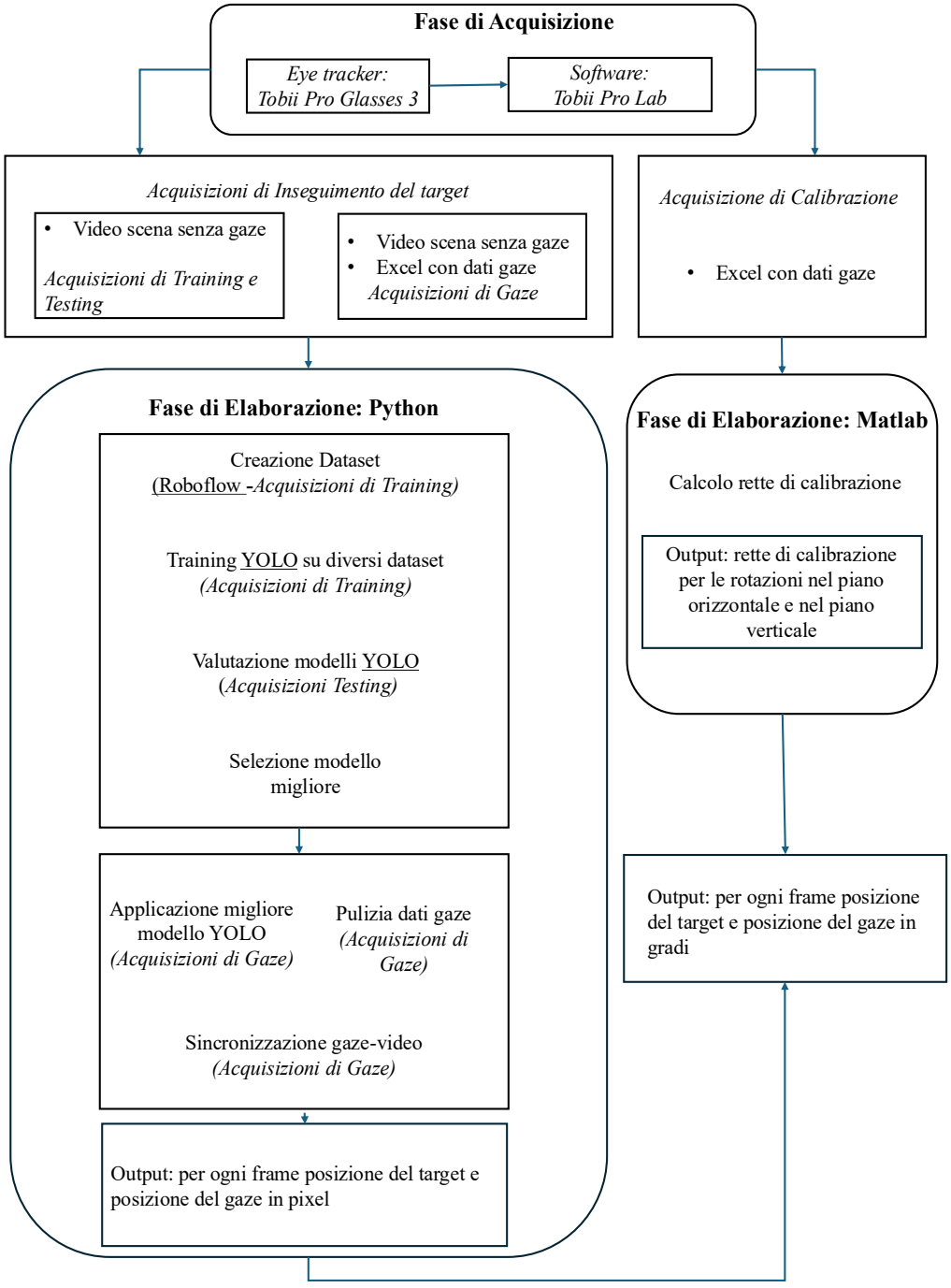

Figura 1: Diagramma di flusso delle fasi di acquisizione ed elaborazione.

Capitolo 1: Stato dell'arte

1.1 Eminattenzione spaziale

L' eminattenzione spaziale è un disturbo neuropsicologico caratterizzato da una ridotta capacità di percepire, orientare l'attenzione o rispondere agli stimoli provenienti dallo spazio controlaterale rispetto alla sede della lesione cerebrale. Tale condizione non è causata da deficit sensoriali o motori di natura primaria, ma da un'alterazione dei meccanismi che regolano l'attenzione e la rappresentazione dello spazio circostante. Può manifestarsi in seguito a diverse patologie neurologiche, tra cui malattie neurodegenerative, tumori cerebrali e traumi cranici; tuttavia, è particolarmente frequente nei pazienti colpiti da ictus cerebrale, raggiungendo nella fase acuta una prevalenza fino all'82%. Sebbene possa essere associata a lesioni di entrambi gli emisferi cerebrali, la sindrome è più frequente e generalmente più grave quando il danno interessa l'emisfero destro. In questi casi, i pazienti tendono a trascurare gli stimoli provenienti dal lato sinistro dello spazio. Questa maggiore incidenza è legata al ruolo predominante dell'emisfero destro nei processi attentivi e nella rappresentazione spaziale. Dal punto di vista neuroanatomico l'eminattenzione spaziale non è associata alla lesione di una singola area cerebrale, ma al coinvolgimento di una rete di regioni che collaborano nel controllo dell'attenzione. In particolare, risultano frequentemente compromessi il network attentivo dorsale (*Dorsal Attention Network*, DAN) e il network attentivo ventrale (*Ventral Attention Network*, VAN), che includono aree parietali, temporali e frontali, soprattutto dell'emisfero destro [1].

Le manifestazioni cliniche possono variare in funzione della modalità sensoriale e della porzione dello spazio coinvolto. Il deficit può infatti coinvolgere la sfera visiva, uditiva o tattile e viene generalmente classificato in tre forme principali in base allo spazio interessato: personale, peripersonale ed extrapersonale. La prima interessa il corpo del paziente; la seconda riguarda lo spazio immediatamente circostante e raggiungibile con gli arti superiori, mentre la terza coinvolge l'ambiente più distante, al di fuori della portata del soggetto.

Le ripercussioni sulla vita quotidiana possono essere significative, riducendo l'autonomia nello svolgimento di attività essenziali quali alimentarsi, leggere, vestirsi, prendersi cura di sé e muoversi in sicurezza nell'ambiente circostante. Il paziente può, ad esempio, trascurare la parte controlesionale del proprio corpo durante le attività di cura personale, consumare soltanto il cibo presente in una metà del piatto, omettere parti di un disegno durante la copia oppure incontrare difficoltà nell'esplorazione dell'ambiente e nell'orientamento spaziale. Per questo motivo, la riabilitazione rappresenta un elemento fondamentale del percorso terapeutico, poiché mira a ridurre gli effetti del disturbo e a favorire il recupero delle funzioni compromesse. Nel corso degli anni sono stati sviluppati numerosi approcci riabilitativi, ciascuno rivolto a specifici aspetti della sindrome [1], [2].

Tra questi, un ruolo particolarmente importante è rivestito dalla riabilitazione visiva, che rappresenta il focus del presente elaborato.

1.2 Approcci tradizionali di riabilitazione visiva

Nonostante i progressi della ricerca, ad oggi non esiste un trattamento per la riabilitazione visiva dell'eminattenzione spaziale universalmente riconosciuto come *gold standard*. La complessità della sindrome e la limitata disponibilità di evidenze scientifiche solide hanno portato allo sviluppo di numerosi approcci riabilitativi, senza che emerga un chiaro consenso su quale sia il più efficace [1]. Le diverse strategie proposte possono essere ricondotte a due principali categorie: approcci *top-down* e approcci *bottom-up*. Sebbene entrambe abbiano l'obiettivo di favorire il recupero dell'orientamento attentivo verso lo spazio controlesionale, differiscono nei meccanismi attraverso cui tale recupero viene promosso. Gli approcci *top-down* si basano sul coinvolgimento attivo del paziente, che deve essere consapevole del proprio deficit e impegnarsi volontariamente nel dirigere l'attenzione verso il lato controlesionale. Per questo motivo, la loro efficacia può risultare limitata nei soggetti affetti da anosognosia, una condizione frequentemente associata all'eminattenzione spaziale che comporta una ridotta o assente consapevolezza del disturbo. Gli approcci *bottom-up*, invece, non

richiedono una partecipazione consapevole da parte del paziente. Essi sfruttano stimolazioni esterne e modificazioni dell'ambiente sensoriale per favorire un riorientamento automatico dell'attenzione verso lo spazio controlesionale. Grazie a questa caratteristica, possono essere applicati efficacemente anche nei soggetti con anosognosia [3]. Tra le strategie riabilitative maggiormente studiate rientrano il *Prism Adaptation Training (PAT)* e lo *Smooth Pursuit Training (SPT)* appartenenti agli approcci *bottom-up*, e il *Visual Scanning Training (VST*) che rientra invece tra gli approcci *top-down* [3], [4].

1.2.1 Prism Adaptation Training (PAT)

Il *Prism Adaptation Training* prevede l'utilizzo di occhiali dotati di lenti prismatiche che inducono una deviazione dell'intero campo visivo verso destra. Durante la procedura standard, il paziente è invitato a eseguire ripetuti movimenti di puntamento verso un bersaglio posto di fronte a lui utilizzando l'arto non affetto. A causa della deviazione visiva indotta dai prismi, i primi tentativi di puntamento risultano sistematicamente spostati verso destra rispetto alla reale posizione del bersaglio. Tuttavia, attraverso la ripetizione del compito, il sistema nervoso rileva progressivamente la discrepanza tra l'informazione visiva percepita e l'esito motorio dell'azione, attivando un processo automatico di ricalibrazione senso-motoria. Questo processo consente al paziente di correggere gradualmente l'errore e di recuperare una maggiore precisione nei movimenti di puntamento. La fase più rilevante del trattamento si verifica al termine dell'esercizio, quando gli occhiali vengono rimossi. In questa fase, la ricalibrazione acquisita dal sistema nervoso persiste temporaneamente, determinando una deviazione dei movimenti verso sinistra, fenomeno noto come *after-effect*. Questo fenomeno favorisce il riorientamento dell'attenzione e dell'azione verso l'emispazio precedentemente trascurato, contribuendo a ridurre i sintomi del neglect [5]. Le principali fasi del *Prism Adaptation Training* sono illustrate in Figura 1.1.

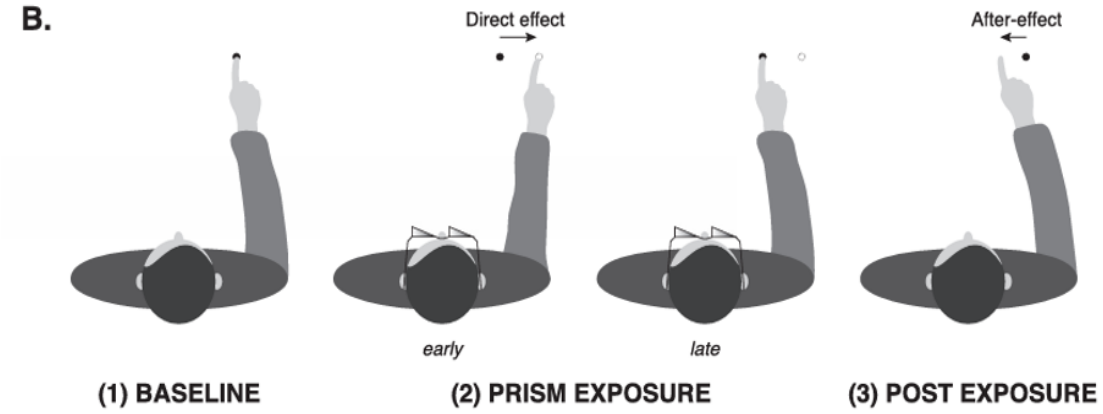

Figura 1.1: Schema del Prism Adaptation Training che mostra la fase di baseline, l'adattamento durante l'esposizione ai prismi e la comparsa dell'after-effect dopo la rimozione delle lenti prismatiche.

1.2.2 Smooth Pursuit Training (SPT)

Lo *Smooth Pursuit Training* impiega stimoli dinamici e richiede al soggetto di seguire visivamente un bersaglio in movimento che si sposta progressivamente verso l'emispazio controlesionale. Questo esercizio coinvolge i movimenti di inseguimento lento (*smooth pursuit*), favorendo l'esplorazione visiva dello spazio attraverso un tracciamento continuo e fluido del bersaglio. È importante che i movimenti di inseguimento siano diretti verso il lato controlesionale, poiché movimenti effettuati nella direzione opposta possono accentuare i sintomi dell'eminattenzione spaziale [3].

1.2.3 Visual Scanning Training (VST)

Il *Visual Scanning Training* si basa sull'esecuzione di compiti di esplorazione visiva mediante l'osservazione di stimoli statici. Le attività proposte comprendono generalmente esercizi carta-e-matita, come test di cancellazione, compiti di ricerca visiva o copia di figure. Un esempio di esercizio di copia di figure è riportato in Figura 1.2. Durante il trattamento, il terapeuta può utilizzare *cue* visivi, verbali o uditivi posizionati sul lato controlesionale per incoraggiare il paziente a iniziare l'esplorazione da quella porzione di spazio. Dal punto di vista oculomotorio, il VST allena principalmente i movimenti saccadici, ovvero i rapidi movimenti oculari che permettono di spostare lo sguardo tra bersagli statici durante l'esplorazione visiva [3].

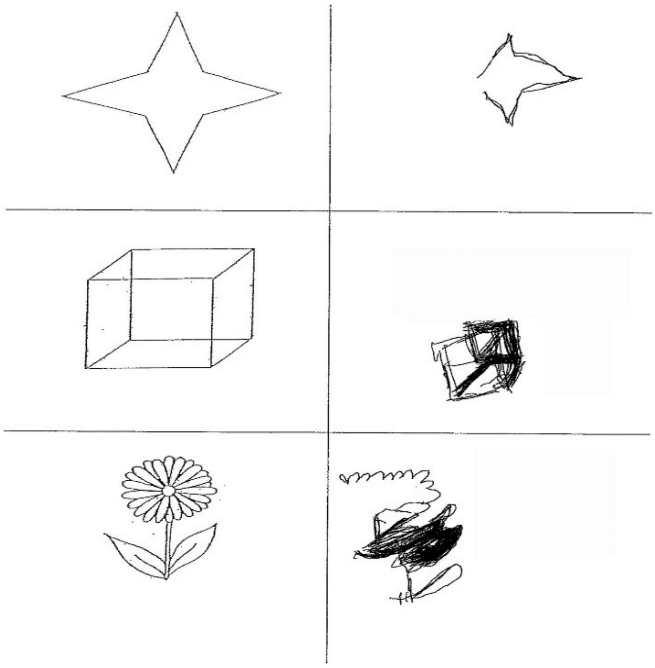

Figura 1.2: Esempio di compito di copia eseguito da un paziente con eminattenzione spaziale, caratterizzato dall'omissione degli elementi localizzati nel lato controlesionale [1].

1.2.4 Limitazioni degli approcci tradizionali

Nonostante l'ampia diffusione di questi approcci, essi presentano alcune limitazioni. Una delle principali riguarda l'assenza di strumenti in grado di monitorare in modo oggettivo e in tempo reale la direzione dello sguardo e l'orientamento della testa durante l'esecuzione degli esercizi. Di conseguenza, il terapeuta non ha accesso diretto al reale comportamento visivo del paziente e deve basare la valutazione principalmente sull'osservazione clinica. Questo rende necessaria una supervisione continua durante l'esercizio, aumentando il carico di lavoro dell'operatore e limitando l'efficienza complessiva dell'intervento riabilitativo. Un'ulteriore limitazione riguarda la natura statica e ripetitiva delle attività proposte, che possono risultare poco coinvolgenti per il paziente [3].

1.3 Tecnologie innovative per la riabilitazione visiva

Per superare alcune delle limitazioni associate agli approcci riabilitativi tradizionali, negli ultimi anni la ricerca si è progressivamente orientata verso l'impiego di soluzioni tecnologiche innovative. Tali strumenti rappresentano una naturale evoluzione delle metodologie cliniche convenzionali e mirano a rendere gli interventi riabilitativi più efficaci, personalizzabili e coinvolgenti per il paziente. In questo contesto, un ruolo centrale è svolto dall'integrazione di tre tecnologie complementari: la Realtà Virtuale (*Virtual Reality*, VR), i sistemi di *Eye Tracking* (ET) e il *biofeedback* [6].

1.3.1 Eye Tracking

L'*Eye Tracking*, o tracciamento oculare, è una tecnologia che consente di rilevare e registrare in modo continuo i movimenti oculari di un soggetto. La sua integrazione negli esercizi riabilitativi permette di ottenere misure quantitative e oggettive del modo in cui il paziente esplora l'ambiente durante l'esecuzione delle attività. Attraverso questa tecnologia è infatti possibile analizzare numerosi parametri oculomotori, tra cui la posizione dello sguardo, l'ampiezza delle saccadi, il numero e la durata delle fissazioni, i tempi di esplorazione e la distribuzione dell'attenzione nello spazio. Ciò consente di comprendere meglio le strategie adottate dal paziente e di monitorarne l'evoluzione nel corso del trattamento riabilitativo [3], [7]. I parametri oculomotori misurati mediante *Eye Tracking* hanno inoltre consentito di caratterizzare il profilo visivo tipico dei pazienti affetti da eminattenzione spaziale. In particolare, i pazienti con neglect sinistro tendono a presentare saccadi di ampiezza ridotta verso l'emispazio controlesionale, un numero inferiore di fissazioni nel campo visivo sinistro e una marcata prevalenza dello sguardo verso il lato destro [7].

La possibilità di quantificare oggettivamente tali alterazioni ha portato all'impiego dell'*Eye Tracking* anche come supporto alla diagnosi. Un esempio significativo è rappresentato dallo studio di *Delazer et al. (2018)*, nel quale pazienti con ictus acuto dell'emisfero destro sono stati sottoposti a compiti di esplorazione visiva

monitorati mediante tracciamento oculare. I risultati hanno evidenziato una marcata deviazione dello sguardo verso destra anche in soggetti che avevano ottenuto punteggi nella norma ai tradizionali test carta e matita. Questo suggerisce che alcuni deficit di esplorazione visiva possano sfuggire alle valutazioni cliniche convenzionali, ma essere identificati attraverso l'analisi dei movimenti oculari. Lo studio evidenzia quindi il potenziale dell'*Eye Tracking* come strumento sensibile per l'identificazione precoce dell'eminattenzione spaziale [8].

1.3.2 Virtual Reality

Un'evoluzione particolarmente promettente riguarda l'integrazione dell'*Eye Tracking* con la Realtà Virtuale. Attraverso ambienti tridimensionali interattivi, spesso fruibili mediante visori *Head-Mounted Display (HMD),* il paziente può svolgere compiti che simulano situazioni della vita quotidiana, come attraversare una strada, cercare oggetti o interagire con elementi in movimento. Questo approccio consente di realizzare esercizi più coinvolgenti e caratterizzati da una maggiore validità ecologica rispetto alle metodiche tradizionali, permettendo di allenare le capacità attentive sia nello spazio peripersonale sia in quello extrapersonale [7].

1.3.3 Biofeedback

L'efficacia dell'integrazione tra Realtà Virtuale ed *Eye Tracking* può essere ulteriormente potenziata mediante l'utilizzo del *biofeedback* in tempo reale. Grazie al monitoraggio continuo della posizione dello sguardo, il sistema è in grado di adattare dinamicamente l'ambiente virtuale in funzione del comportamento del paziente. Ad esempio, quando il soggetto orienta correttamente l'attenzione verso uno stimolo collocato nell'emispazio controlesionale e mantiene la fissazione per il tempo richiesto, il sistema può fornire un *feedback* immediato, visivo, sonoro o multimodale. Un esempio di esercizi riabilitativi che utilizzano questo meccanismo di biofeedback è riportato in Figura 1.3. Questo meccanismo di rinforzo favorisce l'apprendimento, aumenta il coinvolgimento del paziente e riduce la necessità di una supervisione continua da parte del terapista [6].

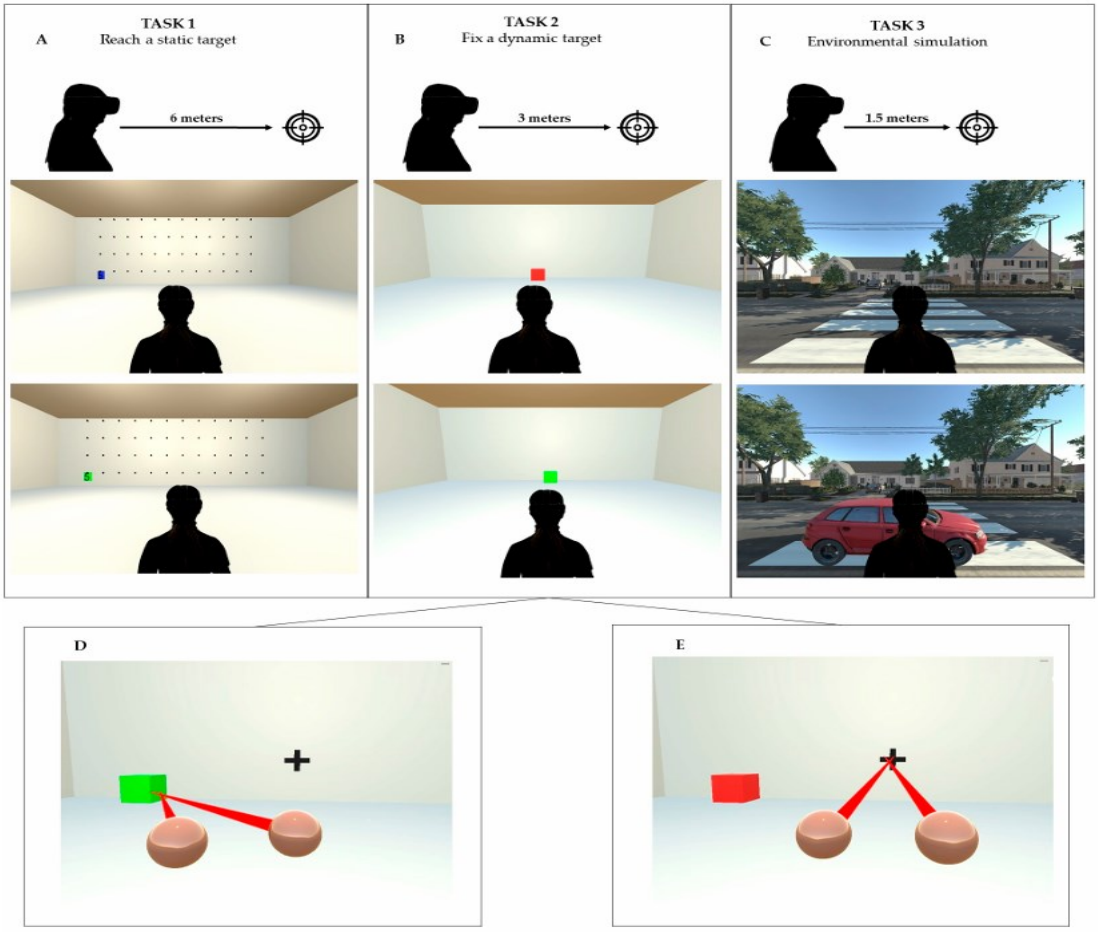

Figura 1.3: Esempi di esercizi riabilitativi in Realtà Virtuale Immersiva basati sull'Eye Tracking. Il paziente è chiamato a fissare o seguire bersagli statici e dinamici; il corretto orientamento dello sguardo viene segnalato tramite un feedback visivo rappresentato dal cambiamento di colore del bersaglio. L'ultimo scenario riproduce un attraversamento pedonale, consentendo l'allenamento delle capacità attentive in un contesto più vicino alle situazioni della vita quotidiana.

1.4 Robotica per la riabilitazione

Negli ultimi anni, la robotica ha assunto un ruolo sempre più importante nella riabilitazione post ictus, grazie alla possibilità di superare alcune limitazioni degli approcci tradizionali. Le terapie convenzionali richiedono infatti un notevole impegno da parte del terapista e non sempre consentono di garantire sessioni sufficientemente intensive e ripetitive. I sistemi robotici permettono invece di erogare esercizi in modo standardizzato e controllato, favorendo allenamenti ad alta intensità e specifici per il compito. In questo contesto, i robot possono essere utilizzati con una duplice funzione. Da un lato, possono agire come dispositivi di assistenza fisica, supportando direttamente il movimento del paziente durante l'esecuzione degli esercizi. Ne sono un esempio gli esoscheletri per l'arto superiore, che guidano e assistono i movimenti di spalla, gomito e polso, consentendo al paziente di svolgere attività riabilitative in modo più efficace e ripetibile. Dall'altro lato, i robot possono assumere il ruolo di veri e propri assistenti intelligenti, secondo il paradigma della *Socially Assistive Robotics* (SAR). In questo caso il loro compito non è quello di muovere fisicamente il paziente, ma di fornire supporto e guida durante il percorso riabilitativo. I moderni robot assistivi possono inoltre riconoscere il paziente, interagire attraverso la voce, monitorare il comportamento durante la seduta e adattare le attività riabilitative in base alle capacità e ai progressi osservati. Di conseguenza, non rappresentano soltanto strumenti per l'esecuzione degli esercizi, ma sistemi in grado di contribuire attivamente alla gestione dell'intero processo terapeutico. Oggi la robotica non si limita più all'assistenza motoria, ma comprende anche funzioni di supervisione, valutazione e supporto cognitivo, contribuendo allo sviluppo di percorsi sempre più personalizzati ed efficaci [9], [10].

1.4.1 TIAGo

Un esempio di robot impiegato nell'ambito dell'assistenza e della riabilitazione è TIAGo, mostrato in Figura 1.4.

Figura 1.4: Robot TIAGo [11].

TIAGo è un robot manipolatore mobile collaborativo sviluppato dall'azienda PAL Robotics [12]. È costituito da una base mobile, un torso regolabile in altezza e un braccio robotico a sette gradi di libertà terminante con una pinza. Nella parte superiore è inoltre presente una testa mobile dotata di telecamera RGB-D, in grado di acquisire informazioni sia di colore sia di profondità dell'ambiente circostante [13]. Il robot è stato utilizzato nell'ambito della *Mission 2 – Activity 5* (*Match Making #53)* del progetto Fit4MedRob *(Fit for Medical Robotics)*, con l'obiettivo di sviluppare e validare un protocollo innovativo per la riabilitazione di pazienti post-ictus affetti da inattenzione emispaziale. A tal fine, sono state svolte attività

di validazione preliminare presso il Living Lab del Centro di Eccellenza REDI dell'Università di Pavia, coinvolgendo soggetti sani.

I test effettuati hanno dimostrato che il robot è in grado di muoversi in modo sicuro nell'ambiente, riconoscere correttamente gli utenti, comprendere i principali comandi vocali e monitorare con elevata precisione l'orientamento della testa del paziente. I risultati ottenuti confermano l'affidabilità della piattaforma robotica, evidenziandone il potenziale impiego in ambito clinico per supportare la riabilitazione di pazienti con deficit neurologici conseguenti all'ictus [14], [15].

Capitolo 2: Strumenti di Acquisizione

2.1 Tobii Pro Glasses 3

Per effettuare le acquisizioni è stato impiegato il sistema Tobii Pro Glasses 3, un *Eye tracker* indossabile progettato per raccogliere dati oculomotori nel mondo reale (Figura 2.1). Il dispositivo opera con una frequenza di campionamento pari a 100 Hz e presenta un'accuratezza di circa 0.6° [16].

Figura 2.1: Tobii Pro Glasses 3 [16] .

2.1.1 Architettura del sistema

Il sistema utilizzato è costituito da tre componenti principali:

- unità testa;
- unità di registrazione portatile;
- applicazione di controllo.

Il dispositivo si presenta come una montatura di occhiali dal design leggero e poco invasivo, progettata per permettere al soggetto di muoversi in modo naturale durante l'acquisizione (Figura 2.2). All'interno della montatura sono integrate

quattro telecamere a infrarossi, due per ciascun occhio, insieme a sedici illuminatori a infrarossi, otto per occhio, posizionati attorno alle lenti.

Questi componenti consentono di illuminare e rilevare con precisione gli occhi del partecipante, permettendo la stima della direzione dello sguardo. Nella parte centrale della montatura è inoltre presente una telecamera di scena che acquisisce il campo visivo frontale del soggetto. La telecamera registra video con una risoluzione di 1920 × 1080 pixel a una frequenza di 25 Hz. In prossimità della telecamera di scena è integrata anche un'unità di misura inerziale (IMU, *Inertial Measurement Unit*), modello ST™ LSM9DS1. Questa unità comprende tre differenti sensori: un accelerometro, un giroscopio e un magnetometro. L'accelerometro e il giroscopio consentono di misurare rispettivamente l'accelerazione lineare e la velocità angolare del capo lungo i tre assi spaziali, mentre il magnetometro viene utilizzato per stimare l'orientamento del dispositivo rispetto al campo magnetico circostante [17], [18].

Figura 2.2: Unità testa [17].

L'unità testa è collegata tramite cavo a un'unità di registrazione portatile esterna, mostrata in Figura 2.3. Quest'ultima rappresenta il modulo di elaborazione del sistema e si occupa della gestione dell'unità testa, oltre che dell'acquisizione e dell'archiviazione dei dati raccolti durante la registrazione. In particolare, i dati di

Eye tracking e i video acquisiti dalla telecamera di scena vengono salvati su una scheda di memoria SD rimovibile. Il sistema è alimentato da una batteria ricaricabile e sostituibile che garantisce un'autonomia di registrazione continua fino a 105 minuti [17].

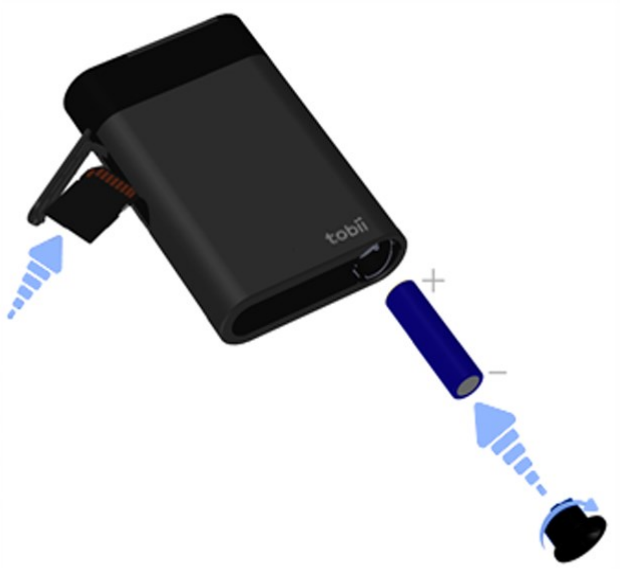

Figura 2.3: Unità di registrazione portatile [17].

La gestione del sistema avviene tramite la Tobii Pro Glasses 3 Controller Application, un software installato su un dispositivo esterno, generalmente un computer. Tramite questa applicazione è possibile controllare le registrazioni, avviandole o interrompendole, e visualizzare in tempo reale i video acquisiti dalla telecamera di scena insieme ai dati di gaze registrati.

La comunicazione tra gli occhiali e il dispositivo di controllo avviene tramite una rete locale, che può essere configurata sia in modalità wireless (Wi-Fi) sia tramite collegamento cablato mediante cavo Ethernet [17], [19]. Le due modalità di connessione sono illustrate in Figura 2.4.

Figura 2.4: Modalità di connessione tra occhiali e dispositivo di controllo [19].

Inoltre, al termine della registrazione, l'applicazione consente di esportare direttamente in formato *.mp4* due differenti tipologie di video:

- il video acquisito dalla telecamera di scena, che mostra l'ambiente osservato dal partecipante;
- lo stesso video con la sovrapposizione grafica del punto di sguardo (*gaze overlay*), rappresentata da un anello rosso che evidenzia il punto verso cui è diretto lo sguardo del partecipante [17], [19].

In Figura 2.5 si può vedere un esempio di video di scena con e senza la sovrapposizione del punto di sguardo.

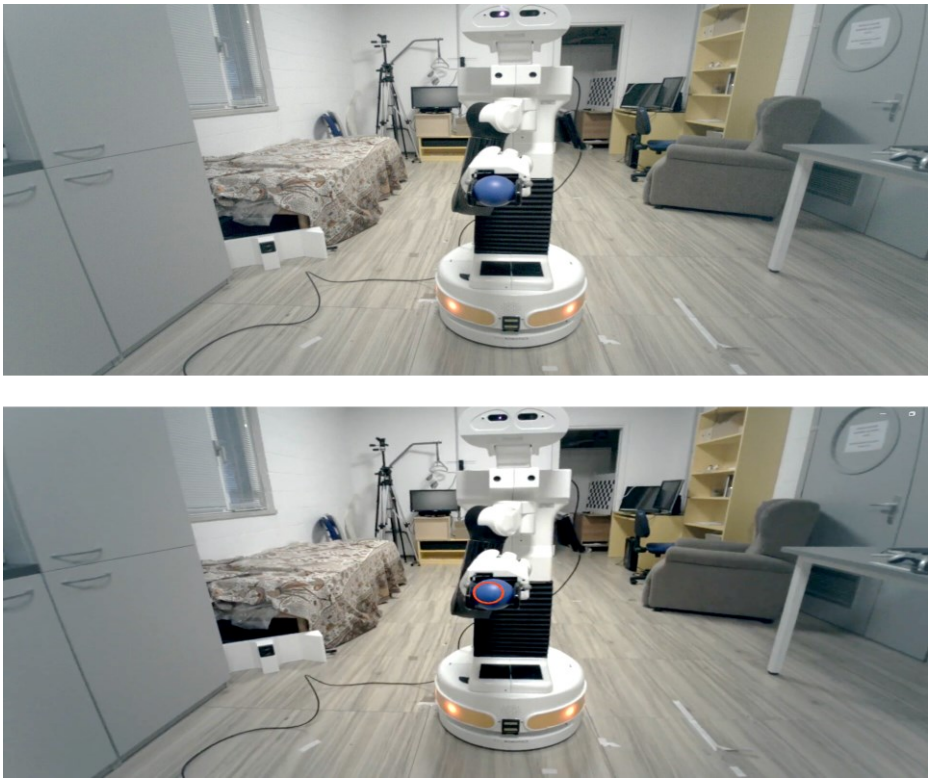

Figura 2.5: Esempio dello stesso frame del video di scena visualizzato senza e con gaze overlay.

2.1.2 Principio di funzionamento

Per acquisire la posizione dello sguardo, il sistema di *Eye tracking* utilizza una versione avanzata e brevettata (US Patent US7,572,008**)** della tecnica PCCR (*Pupil-Center Corneal Reflection*) [20], illustrata in Figura 2.6.

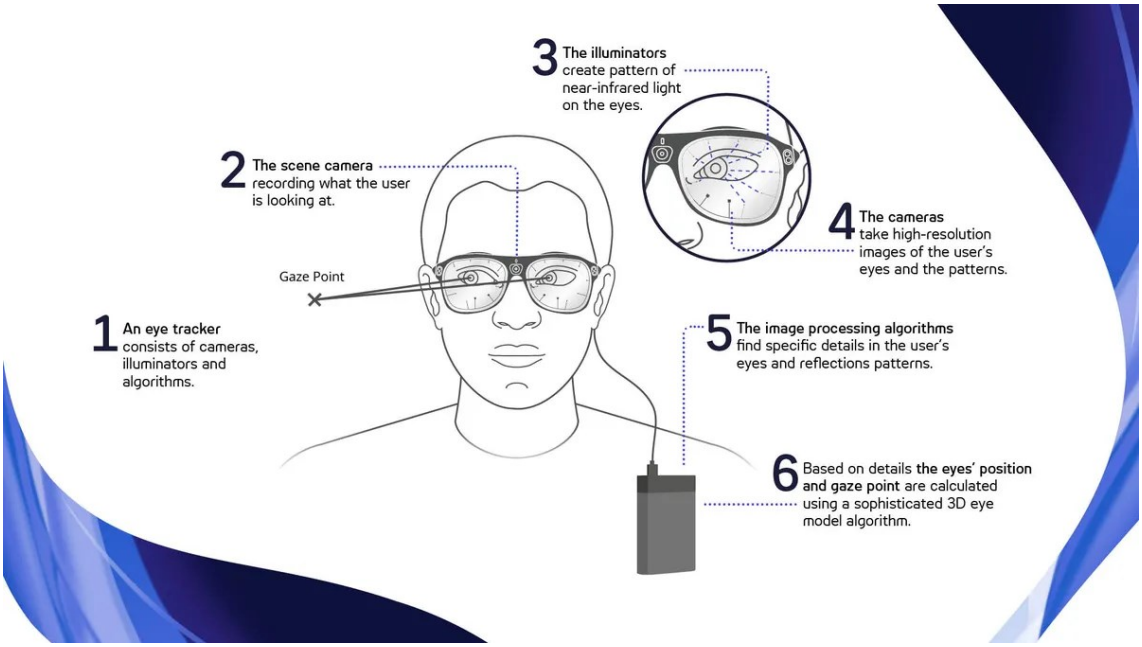

Figura 2.6: Principio di funzionamento della tecnica utilizzata [21].

Il principio di base di questa tecnica consiste nell'illuminare gli occhi mediante illuminatori integrati nella montatura degli occhiali, che emettono luce nel vicino infrarosso, e nell'acquisire le immagini oculari tramite le telecamere anch'esse integrate direttamente nel dispositivo. L'utilizzo della luce infrarossa presenta diversi vantaggi: non è percepita dall'occhio umano e quindi non provoca fastidio o distrazioni durante l'acquisizione; inoltre risulta meno influenzata dalle variazioni di illuminazione ambientale rispetto alla luce visibile, permettendo di ottenere immagini oculari più stabili e misurazioni più affidabili anche in ambienti differenti [22]. All'interno delle immagini oculari acquisite, gli algoritmi di elaborazione individuano due elementi principali: il centro della pupilla e i riflessi luminosi generati sulla superficie della cornea dagli illuminatori ad infrarossi, chiamati *glint*.

Il funzionamento della tecnica PCCR si basa sul fatto che, quando il soggetto sposta lo sguardo, la posizione della pupilla varia rispetto ai riflessi corneali, che rimangono invece quasi stabili. Analizzando continuamente la posizione relativa tra il centro pupillare e i *glint*, il sistema riesce a stimare la direzione dello sguardo e quindi il punto osservato dal soggetto [22].

La relazione tra il centro pupillare e i riflessi corneali è illustrata in Figura 2.7.

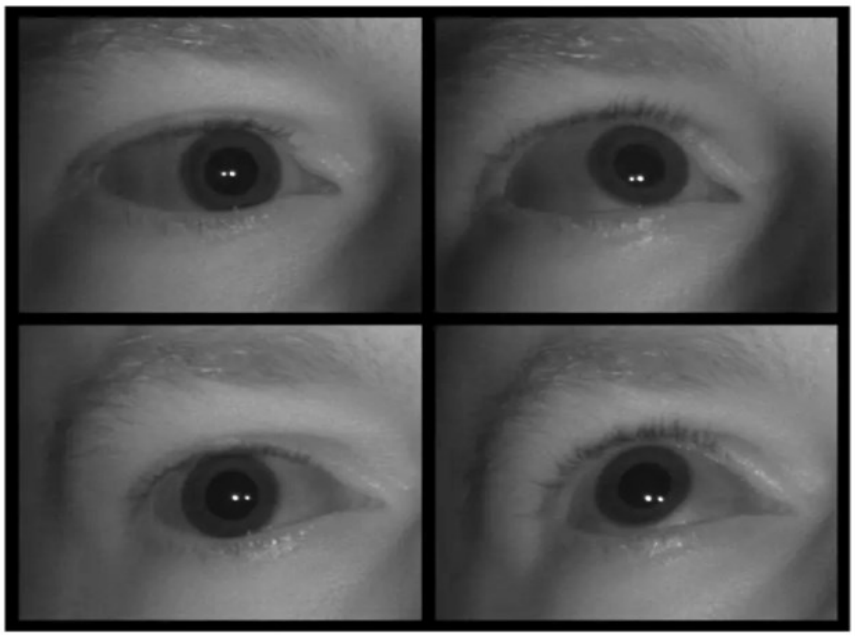

Figura 2.7: Relazione tra centro pupillare e riflessi corneali [22].

Rispetto al metodo PCCR tradizionale, la tecnologia sviluppata da Tobii introduce un sistema di illuminazione infrarossa dinamico che alterna rapidamente due differenti configurazioni luminose (Figura 2.8).

La prima modalità utilizza illuminatori coassiali e produce il cosiddetto effetto *bright pupil*, nel quale la pupilla appare particolarmente luminosa e quindi più semplice da identificare. Tuttavia, questa configurazione riduce il contrasto dell'immagine e rende più difficile rilevare con precisione i riflessi corneali. La seconda modalità utilizza invece illuminatori esterni e genera l'effetto *dark pupil,* che permette di rilevare con maggiore precisione i riflessi corneali periferici utili per stimare la posizione tridimensionale dell'occhio.

L’utilizzo di una sola configurazione luminosa per stimare contemporaneamente sia la direzione dello sguardo sia la posizione spaziale dell’occhio potrebbe ridurre l’accuratezza complessiva del sistema. Per questo motivo il dispositivo alterna rapidamente le due modalità di illuminazione, combinando le informazioni ottenute da entrambe [20].

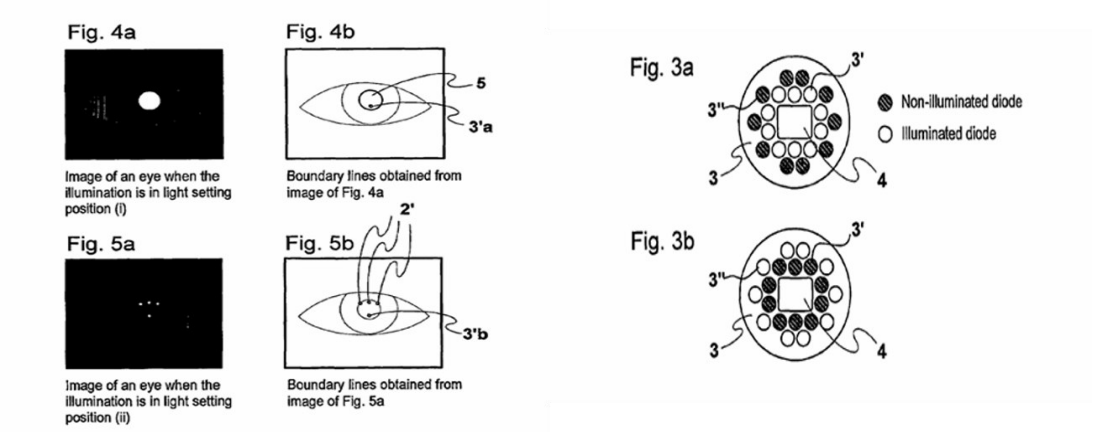

Figura 2.8: A sinistra sono mostrati gli effetti prodotti sull’immagine oculare nelle modalità bright pupil e dark pupil; a destra le corrispondenti configurazioni di accensione dei LED infrarossi utilizzate per ottenere le due modalità di illuminazione [20].

L’integrazione delle informazioni provenienti dalle due modalità consente agli algoritmi di elaborazione di costruire un modello tridimensionale dell’occhio più accurato e di stimare il punto di sguardo anche in presenza di movimenti della testa durante l’acquisizione [21], [22].

2.1.3 Calibrazione a 1 punto

Per poter stimare correttamente il punto di sguardo utilizzando il modello tridimensionale dell’occhio descritto in precedenza, il sistema richiede una procedura di calibrazione eseguita prima di ogni registrazione. Viene quindi effettuata una calibrazione a un punto (*One-Point Calibration*). Poiché ogni individuo presenta caratteristiche oculari differenti, come la forma della cornea e la posizione della fovea, il modello tridimensionale dell’occhio utilizzato dal sistema deve essere adattato alle specifiche caratteristiche anatomiche del partecipante.

Questa procedura consente quindi di personalizzare gli algoritmi di stima dello sguardo sulla base della geometria oculare individuale. La calibrazione viene eseguita tramite la Tobii Pro Glasses 3 Controller Application. Prima dell'inizio di ogni registrazione, il partecipante indossa gli occhiali e si posiziona frontalmente rispetto a una *calibration card*, mantenendola a una distanza compresa tra 50 e 100 cm [19].

La Figura 2.9 mostra la *calibration card* utilizzata.

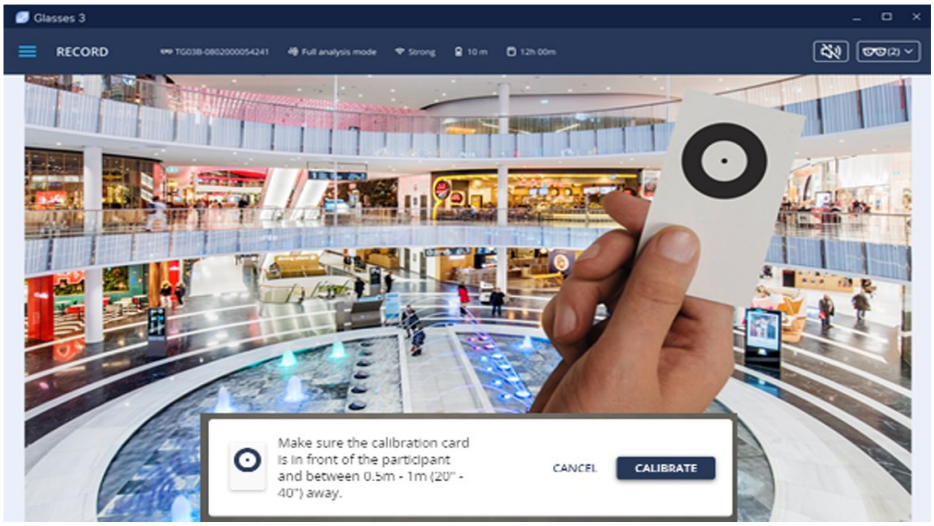

Figura: 2.9: Calibration Card [19].

Durante questa fase, il sistema rileva automaticamente il marcatore presente sulla *card* e acquisisce i campioni di sguardo necessari per costruire un modello tridimensionale personalizzato dell'occhio del soggetto. In presenza di evidenti disallineamenti (*offset*), è consigliabile ripetere la procedura di calibrazione per garantire l'affidabilità delle acquisizioni. In alternativa alla scheda manuale, è possibile utilizzare una *calibration card* adesiva applicata direttamente a una superficie fissa dell'ambiente [19], [23].

2.2 Tobii Pro Lab

Come illustrato precedentemente, tramite l'applicazione di controllo di Tobii Pro Glasses 3 è possibile esportare il video della scena, sia con sia senza la sovrapposizione del punto di sguardo [19], [24] .

Per ottenere invece i dati numerici relativi alla posizione dello sguardo all'interno del video è necessario utilizzare il software Tobii Pro Lab. In particolare, i dati salvati sulla scheda SD presente nell'unità di registrazione portatile vengono importati all'interno di un *Glasses Project* creato nel software. Questa tipologia di progetto è creata specificamente per la gestione, la visualizzazione e l'analisi dei dati acquisiti tramite i sistemi di *Eye tracking* Tobii Pro Glasses [19], [24].

La finestra di creazione di un nuovo Glasses Project è mostrata in Figura 2.10

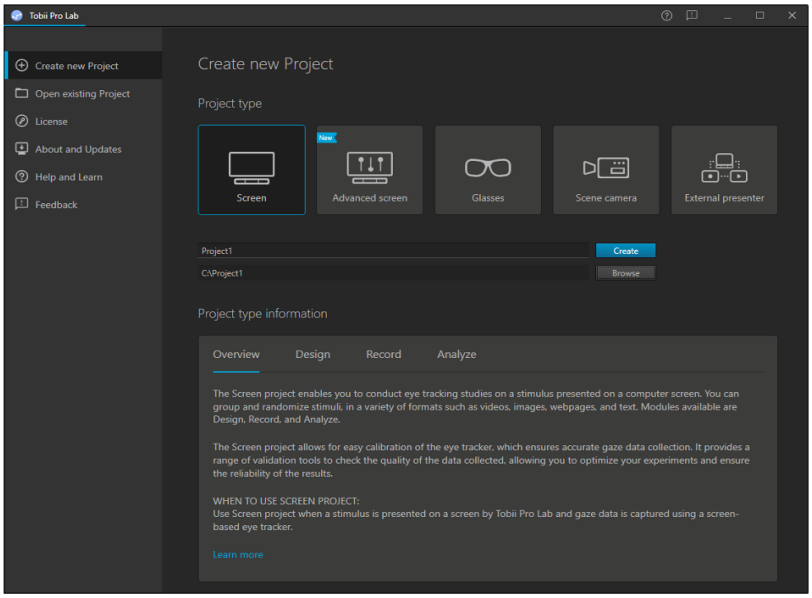

Figura 2.10: Creazione di un Glasses Project in Tobii Pro Lab [24].

2.2.1 Modulo Analyze: TOI e Gaze Filter

Una volta importati i dati all'interno del progetto, accedendo al modulo *Analyze* è possibile visualizzare le registrazioni video con la sovrapposizione del punto di sguardo. Questo modulo mette a disposizione diversi strumenti per l'elaborazione dei dati acquisiti. Nel presente lavoro sono state utilizzate principalmente due operazioni.

La prima operazione riguarda la definizione dei *Times of Interest* (TOI). Poiché le acquisizioni continue includono inevitabilmente anche fasi precedenti o successive al compito sperimentale, i TOI consentono di selezionare esclusivamente il segmento temporale rilevante per l'analisi. Per definire questi intervalli viene utilizzata la funzione *Custom Event Types*, che consente di inserire manualmente lungo la timeline della registrazione un evento di inizio e un evento di fine. Successivamente, tramite la funzione *Time of Interest*, viene creato l'intervallo temporale compreso tra i due eventi precedentemente definiti [24].

Le sezioni *Custom Event Types* e *Times of Interest* del software Tobii Pro Lab sono mostrate in Figura 2.11.

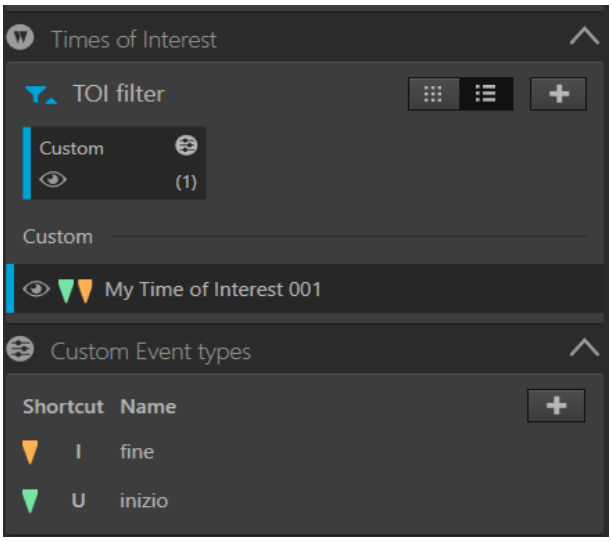

Figura 2.11: Definizione dei Times of Interest (TOI).

La seconda operazione utilizzata riguarda la scelta del *Gaze Filter*, ovvero il filtro applicato ai dati acquisiti dall'*Eye tracker*. Il *software* mette a disposizione diversi filtri predefiniti: *Raw* e *I-VT (Identification by Velocity Threshold) Attention* e *Fixation*. Il filtro *Raw Gaze* non applica alcuna elaborazione ai campioni registrati,

mentre i filtri *I-VT* dopo una prima fase di elaborazione, assegnano a ciascun dato una velocità angolare e ne effettuano una classificazione sulla base di soglie di velocità predefinite. In particolare, i campioni con velocità inferiore alla soglia vengono identificati come fissazioni, mentre quelli con velocità superiore vengono classificati come saccadi [24].

Nel presente lavoro, la scelta del filtro è stata effettuata in funzione della tipologia di acquisizione e dell'obiettivo dell'analisi.

Per le acquisizioni di inseguimento del target è stato scelto il filtro *Raw Gaze,* un'opzione che permette di mantenere inalterati i dati acquisiti.

Le impostazioni predefinite del filtro utilizzato sono riportate nella Figura 2.12.

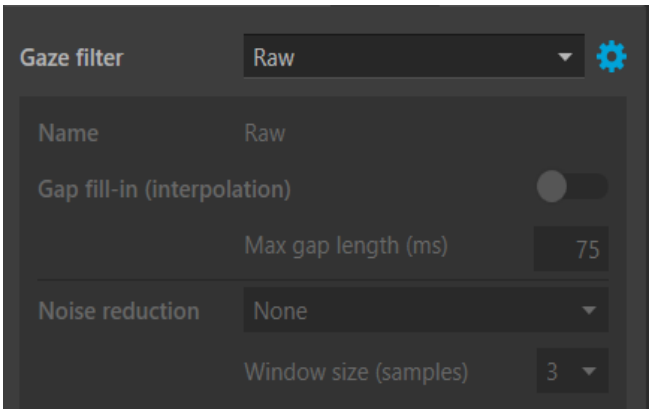

Figura 2.12: Impostazioni del filtro Raw Gaze.

Come mostrato nella Figura 2.12, le opzioni *Gap fill-in* e *Noise Reduction* risultano disattivate, in questo modo viene garantito l'accesso ai dati grezzi non alterati. Tra le impostazioni è inoltre presente la funzione *Eye Selection*, impostata su *Average*. Con questa configurazione, se in un determinato istante il sistema riesce a tracciare correttamente un solo occhio, il punto di sguardo binoculare viene comunque stimato utilizzando l'unico dato valido disponibile [25].
Per le acquisizioni di calibrazione è stato invece selezionato il filtro *I-VT Fixation*. Questo filtro è progettato per studi svolti in condizioni controllate e caratterizzati da movimenti minimi del capo. Le acquisizioni di calibrazione rientrano proprio in questa categoria, poiché durante il task il soggetto mantiene la testa ferma

mentre osserva i punti del diagramma di calibrazione. Il filtro classifica i movimenti oculari in base alla velocità con cui varia la direzione dello sguardo. In particolare, utilizza una soglia di velocità pari a 30°/s: i movimenti con velocità inferiore vengono classificati come fissazioni (*fixations*), mentre quelli superiori come saccadi (*saccades*).

Le impostazioni predefinite del filtro sono riportate nella Figura 2.13.

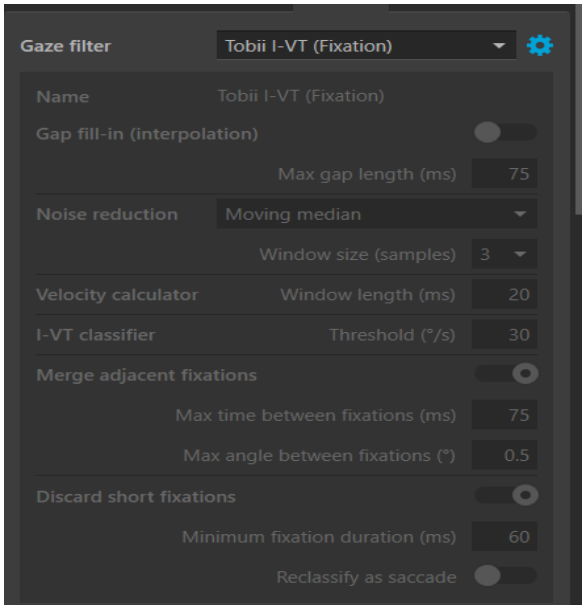

Figura 2.13: Impostazioni del filtro I-VT (Fixation).

Come nel filtro *Raw Gaze*, anche in questo caso la funzione *Gap fill-in* risulta disattivata e la funzione *Eye selection* è impostata su *Average*. In questo filtro è invece attiva la funzione *Noise Reduction*, impostata su *Moving Median*. Questa funzione serve a ridurre il rumore presente nei dati acquisiti, dovuto sia a interferenze ambientali sia a piccoli movimenti oculari involontari, come tremori e microsaccadi. L'algoritmo analizza piccoli gruppi consecutivi di campioni, definiti dal parametro *Window Size* (impostato a tre campioni), e sostituisce ciascun gruppo con il valore mediano dei campioni presenti nella finestra temporale considerata. In questo modo il segnale risulta più stabile e meno soggetto a fluttuazioni indesiderate.

L'elemento principale del filtro riguarda la classificazione dei movimenti oculari sulla base della velocità con cui varia la direzione dello sguardo. Questa

operazione viene eseguita dalla funzione *Velocity Calculator*, che assegna a ciascun campione una velocità espressa in gradi al secondo (°/s).

Per il calcolo della velocità, il *software* utilizza una finestra temporale mobile (*sliding window*). Nel caso in cui siano presenti campioni mancanti all'interno della finestra, il sistema non è in grado di calcolare la velocità. Questa condizione si verifica tipicamente durante i *blink* oppure nelle porzioni iniziali e finali della registrazione. Di conseguenza, l'applicazione del filtro può aumentare leggermente il numero di dati mancanti rispetto al segnale grezzo originale.

Il filtro applica anche una soglia, *Velocity Threshold*, impostata di default a 30°/s. I campioni con velocità inferiore alla soglia vengono classificati come fissazioni, mentre quelli con velocità uguale o superiore vengono classificati come saccadi. Nei casi in cui la velocità non possa essere calcolata, il movimento viene classificato come *Unknown Eye Movement*.

Per migliorare ulteriormente la classificazione, il filtro integra alcune funzioni aggiuntive. La funzione *Merge Adjacent Fixations* permette di unire fissazioni molto vicine tra loro che potrebbero essere state separate erroneamente a causa del rumore del segnale. La funzione *Discard Short Fixations*, invece, elimina le fissazioni troppo brevi per essere considerate rappresentative di una reale attenzione visiva. Questa operazione si basa sul parametro *Minimum Fixation Duration*. Gli eventi con durata inferiore a tale soglia vengono quindi riclassificati come *Unknown Events*. Risulta infine disattivata l'opzione *Reclassify as Saccade*, che consentirebbe di unire due saccadi separate da un breve intervallo classificato come *Unknown* [25], [26].

2.2.2 Export dei dati

Una volta definite le modalità di pre-elaborazione dei dati, il passaggio successivo consiste nell'esportazione del file contenente le informazioni relative alla posizione dello sguardo. Questa operazione viene eseguita tramite la funzione *Data Export*, disponibile all'interno del modulo *Analyze* [24]. Durante l'esportazione è possibile definire diversi parametri che determinano la struttura e il contenuto del file generato.

Le impostazioni selezionate per l'acquisizione di calibrazione e per le acquisizioni di inseguimento del target sono riportate in Figura 2.14.

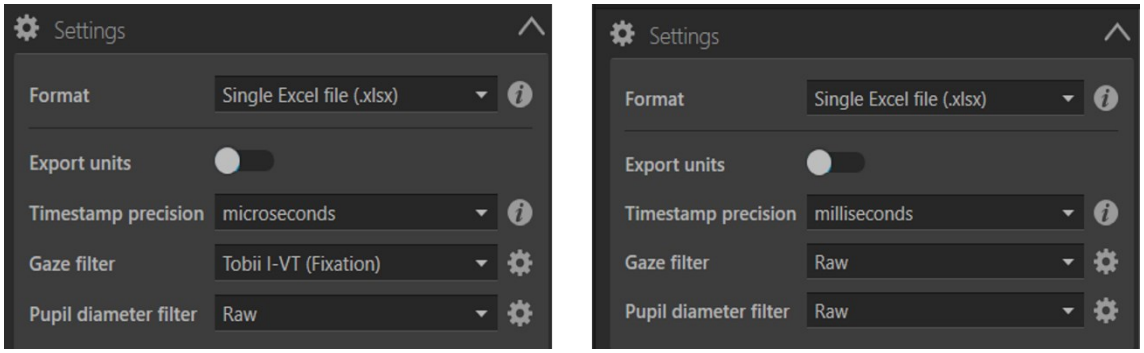

Figura 2.14: Configurazione dell'esportazione dei dati per l'acquisizione di calibrazione (sinistra) e per le acquisizioni di inseguimento del target (destra).

Per entrambe le tipologie di acquisizioni è stato selezionato il formato compatibile con Excel così da semplificare la successiva lettura e analisi dei dati.

Per quanto riguarda il parametro *Timestamp precision*, che definisce l'unità di misura utilizzata nella linea temporale, nelle acquisizioni di calibrazione è stata utilizzata una risoluzione in microsecondi, così da mantenere un livello di precisione temporale più elevato. Per le acquisizioni di inseguimento del target è stata invece scelta una risoluzione in millisecondi, ritenuta sufficiente per le elaborazioni successive e utile per ottenere dataset più snelli e semplici da gestire.

Durante l'esportazione viene inoltre confermato il *Gaze Filter* selezionato in precedenza e viene definito il *Pupil Diameter Filter* che per entrambi è stato impostato a *Raw.*

Il processo di esportazione genera un file Excel organizzato in colonne, contenente numerose informazioni che possono essere suddivise in tre principali categorie:

- Informazioni generali che comprendono i metadati necessari per identificare e contestualizzare temporalmente la registrazione. Tra questi, il parametro più rilevante per il presente lavoro è il *Recording Timestamp*, ovvero il riferimento temporale associato a ciascun campione acquisito dall'*Eye tracker [27]*;
- Dati di *Eye tracking* che comprendono le informazioni relative alla posizione dello sguardo. In particolare:
    - *Gaze point 2D:* coordinate dello sguardo espresse in pixel rispetto al frame video;
    - *Gaze point 3D:* punto di vergenza dello sguardo nello spazio tridimensionale;
    - *Gaze direction:* vettore direzionale dello sguardo per ciascun occhio;
    - *Pupil position e Pupil diameter:* posizione tridimensionale e diametro della pupilla;
    - *Validity of eye data*: parametro che indica se il sistema è riuscito a rilevare correttamente gli occhi nel dato istante temporale
- Dati provenienti da altri sensori cioè le informazioni registrate da giroscopio, accelerometro e magnetometro integrati nel dispositivo [24], [25].

Tra tutti i parametri disponibili, per le successive elaborazioni sono state utilizzate principalmente le variabili *RecordingTimestamp*, *GazePointX* e *GazePointY*.

La scelta di utilizzare *GazePointX* e *GazePointY* rispetto agli altri parametri relativi allo sguardo è legata al fatto che, ai fini di questo lavoro, era necessario disporre di un punto di *gaze* binoculare espresso direttamente in pixel. Le coordinate sono infatti espresse rispetto a un sistema di riferimento associato al frame del video di scena, con origine nell'angolo superiore sinistro dell'immagine: l'asse X aumenta verso destra, mentre l'asse Y aumenta verso il basso [9].

Capitolo 3: Procedura di Calibrazione

L'obiettivo finale di questa fase di elaborazione è la costruzione di un modello di calibrazione basato sulla regressione lineare. Lo scopo del modello è descrivere nel modo più accurato possibile la relazione tra i dati acquisiti dall'*Eye tracker, Gaze Point X e GazePoint Y* espressi in pixel (x, y), e i corrispondenti angoli reali di movimento oculare (θ_x, θ_y). Poiché le coordinate *GazePointX* e *GazePointY* sono ottenute dalla combinazione delle informazioni provenienti da entrambi gli occhi, la procedura di calibrazione viene eseguita in modalità binoculare.

In particolare, vengono ricavate due rette di regressione separate, una relativa all'asse orizzontale X e una relativa all'asse verticale Y. Le due relazioni possono essere descritte tramite le seguenti equazioni:

$$\theta_x = a_x x + b_x$$

$$\theta_y = a_y y + b_y$$

dove θ_x *e* θ_y rappresentano gli angoli oculari reali, mentre x e y indicano le coordinate della posizione dello sguardo acquisite dall'*Eye tracker*. I parametri a e b rappresentano rispettivamente il coefficiente angolare e l'intercetta della retta.

La stima di tali parametri consente quindi di convertire le coordinate espresse in pixel nei corrispondenti angoli oculari, trasformando i dati grezzi acquisiti dall'*Eye tracker* in misure del movimento dell'occhio [28].

3.1 Griglia di calibrazione

Per eseguire la procedura di calibrazione è stata realizzata una griglia di calibrazione costituita da 9 punti disposti su tre righe e tre colonne, configurazione che rappresenta uno degli standard più utilizzati per ottenere una mappatura accurata dello sguardo [28]. Nella fase iniziale di validazione il pannello della griglia veniva posizionato a muro (Figura 3.1).

Figura 3.1: Griglia di calibrazione posizionata a muro.

Successivamente, nella configurazione finale, la griglia è stata montata su un supporto e mantenuta manualmente sopra il target utilizzato durante le acquisizioni di inseguimento (Figura 3.2). Questa scelta è stata adottata poiché, come descritto in precedenza, l'acquisizione di calibrazione e quella di inseguimento devono essere eseguite consecutivamente senza rimuovere gli occhiali di *Eye tracking;* tuttavia, rappresenta certamente una fonte di errori per la procedura.

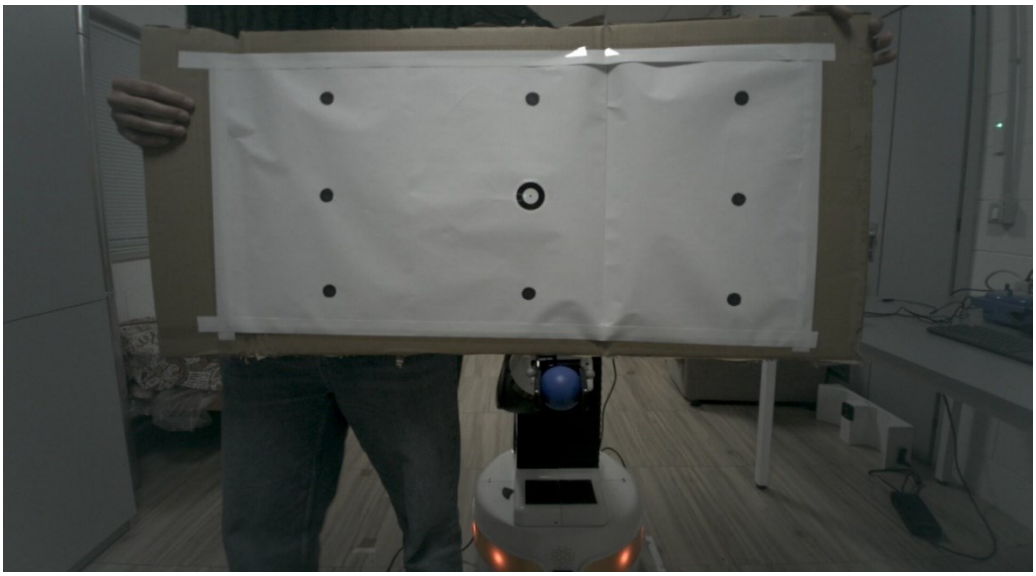

Figura 3.2: Configurazione finale del diagramma di calibrazione.

Di seguito vengono descritte le principali caratteristiche geometriche della griglia utilizzata. La distanza tra il soggetto e il pannello è stata fissata a 85 cm, corrispondente alla distanza utilizzata durante le acquisizioni finali di inseguimento visivo. Sulla base di questa, sono state calcolate le distanze tra i punti della griglia in modo da ottenere eccentricità visive di circa 20° lungo l'asse orizzontale e 10° lungo l'asse verticale rispetto ad un occhio "ciclopico" posto sul punto medio del segmento che unisce i due occhi.

Il calcolo degli angoli di rotazione necessari perché l'occhio ciclopico possa fissare i diversi target della griglia è stato effettuato mediante la seguente relazione:

$$\vartheta = \tan^{-1}(\frac{d}{D}) \quad (1)$$

dove ϑ rappresenta l'angolo visivo espresso in gradi, d indica la distanza fra i punti della griglia in cm e D rappresenta la distanza fra il soggetto e il diagramma e in cm [29].

Sulla base di tale calcolo, le distanze tra i punti sono risultate pari a 31 cm lungo l'asse orizzontale e 15 cm lungo l'asse verticale. I punti del diagramma presentano un diametro di 2 cm che, alla distanza di osservazione utilizzata, corrisponde a un angolo visivo di circa 1°, calcolato mediante la relazione precedentemente descritta (1) [29].

La griglia viene posizionata in modo che il punto centrale risulti allineato con il punto medio del segmento che unisce gli occhi del soggetto, ovvero la posizione dell'occhio ciclopico. Questo allineamento definisce l'origine geometrica del sistema di riferimento spaziale, considerando positivi i valori verso destra e verso l'alto [29]. Infine, nel punto centrale del pannello è stato applicato l'adesivo di calibrazione.

Durante la fase di acquisizione il soggetto riceve istruzioni vocali che indicano il punto del diagramma da fissare. Per ciascun punto è stato mantenuto un tempo di

fissazione pari a cinque secondi, così da raccogliere un numero sufficiente di campioni senza prolungare eccessivamente la durata della procedura.

La sequenza di fissazione è stata definita nel seguente modo: il soggetto inizia osservando il punto centrale e successivamente osserva, in senso orario, tutti i punti disposti lungo il contorno della griglia. Dopo ogni fissazione periferica, il soggetto torna sempre a fissare il punto centrale. Durante l'acquisizione è importante che il soggetto mantenga la testa il più possibile ferma. Infatti, eventuali movimenti del capo influenzerebbero la stima degli angoli oculari, rendendo necessario considerare anche il contributo della rotazione della testa nel calcolo finale. Generalmente, i movimenti dello sguardo vengono eseguiti principalmente tramite i soli movimenti oculari fino a eccentricità di circa ±15° [29]. Nel presente studio, però, il diagramma di calibrazione è stato progettato in modo da raggiungere eccentricità orizzontali fino a circa 20°; per questo motivo al soggetto è stato richiesto di mantenere volontariamente la testa il più immobile possibile durante tutta la procedura.

3.2 Elaborazione dei dati

Per l'elaborazione dell'acquisizione di calibrazione è stata sviluppata una pipeline automatizzata in Matlab composta da diverse funzioni. L'intera procedura può essere suddivisa nei seguenti passaggi principali:

- lettura del file Excel esportato da Tobii Pro Lab*;*
- rappresentazione grafica dei segnali *GazePointX e GazePointY* rispetto al *Recording TimeStamp;*
- calcolo dei valori mediani di *GazePointX e GazePointY* per ciascun punto di fissazione;
- calcolo delle rette di calibrazione per gli assi X e Y.

3.2.1 Importazione e preprocessing

Come prima operazione, il file Excel viene importato in Matlab mediante la funzione *readtable.* Poiché il dataset contiene un numero elevato di informazioni non necessarie ai fini dell'analisi, viene eseguita una fase di *preprocessing* durante la quale:

- vengono mantenute solo le righe relative all'*Eye tracker*, eliminando quindi le informazioni provenienti dai sensori IMU;
- vengono selezionate solamente le colonne *GazePointX* e *GazePointY* e *RecordingTimestamp*.

Nella Tabella 3.1 sono riportate le prime righe del dataset ottenuto dopo questa fase di pulizia.

Tabella 3. 1: Tabella ottenuta dopo la fase di preprocessing.

	1 RecordingTimestamp	2 GazePointX	3 GazePointY
1	5149781	'989'	'377'
2	5159767	'989'	'377'
3	5169831	'989'	'377'
4	5179858	'989'	'377'
5	5189854	'989'	'377'
6	5199839	'988'	'378'
7	5209903	'988'	'378'
8	5219930	'988'	'378'
9	5229927	'987'	'378'
10	5239912	'987'	'378'
11	5249977	'987'	'378'
12	5260003	'986'	'377'
13	5269999	'986'	'377'

3.2.2 Calcolo delle coordinate mediane

Successivamente vengono rappresentati graficamente i dati *GazePointX* e *GazePointY* in funzione del *RecordingTimestamp*. Il grafico risultante presenta un andamento a gradini, in cui ciascun gradino corrisponde a una fase di fissazione di uno specifico punto del diagramma di calibrazione. Conoscendo l'ordine con cui i punti vengono osservati durante il task sperimentale, è quindi possibile associare ciascun gradino al corrispondente punto della griglia. Come mostrato nella Figura 3.3, in alcune porzioni del segnale sono presenti discontinuità dovute agli istanti temporali in cui *l'Eye tracker* non è riuscito a rilevare correttamente la posizione dell'occhio. Nel segnale si osservano complessivamente 16 gradini, poiché il punto centrale del diagramma viene fissato più volte durante la procedura sperimentale, in particolare otto volte.

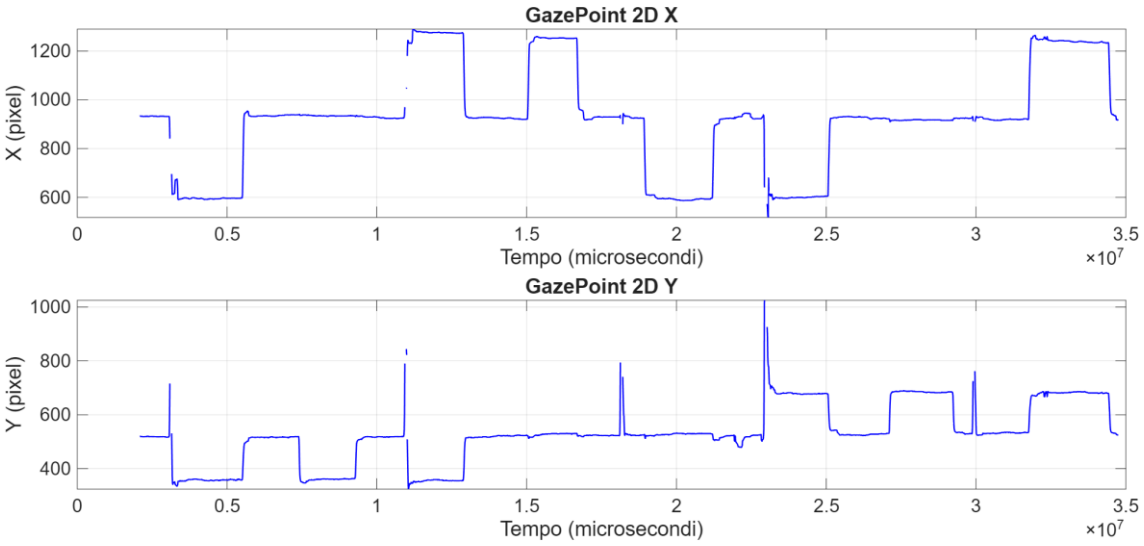

Figura 3.3: Andamento temporale dei segnali GazePointX e GazePointY.

A partire da questi grafici vengono successivamente calcolati i valori mediani di *GazePointX* e *GazePointY* per ciascun gradino del segnale (Figura 3.4). Questa operazione viene eseguita mediante le funzioni *linkaxes* e *ginput(2)*.

In particolare, la funzione *ginput(2)* consente di selezionare manualmente, per ogni gradino, un punto iniziale e un punto finale, definendo così l'intervallo temporale utilizzato per il calcolo della mediana.

Durante questa selezione viene lasciato un piccolo intervallo iniziale dall'inizio del gradino, poiché il soggetto necessita di un breve tempo per spostare e stabilizzare lo sguardo sul nuovo target [29].

Per garantire che l'intervallo selezionato sia identico sia per il segnale *GazePointX* sia per *GazePointY*, viene utilizzata la funzione *linkaxes*, che sincronizza gli assi dei due grafici. Di conseguenza, le selezioni effettuate su un grafico vengono automaticamente riportate anche sull'altro.

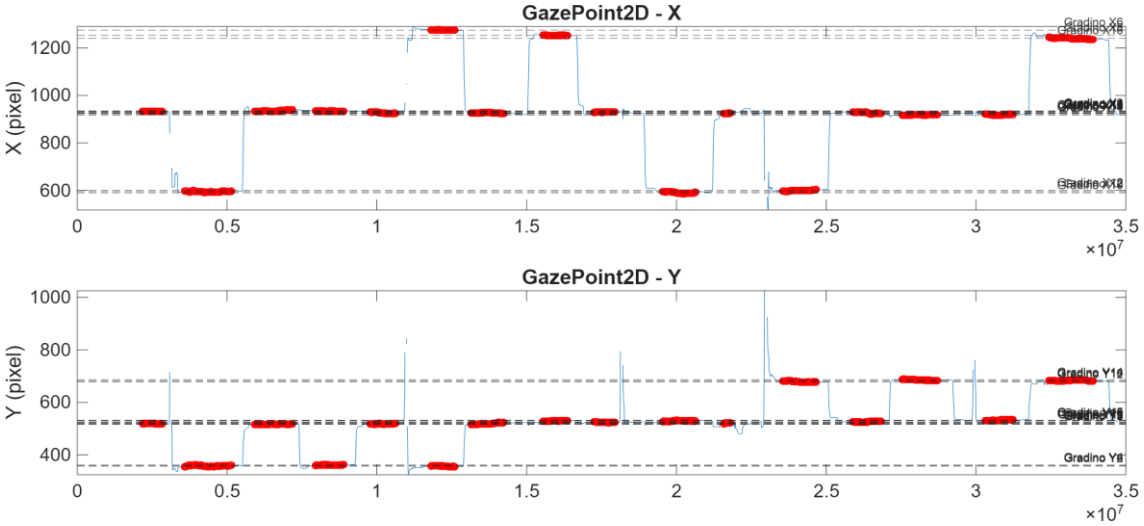

Figura 3.4: Selezione degli intervalli temporali utilizzati per il calcolo delle mediane.

Al termine della procedura si ottiene, per ciascun punto del diagramma un valore mediano di *GazePointX* e *GazePointY* espresso in pixel.

Tali valori sono riportati nella Tabella 3.2.

Tabella 3.2: Valori mediani di GazePointX e GazePointY per ciascun punto del diagramma.

	Mediana Gaze X [pixel]	**Mediana Gaze Y [pixel]**
Punto centrale 1	933	519
Punto in Alto Sinistra	595	358
Punto centrale 2	935	516
Punto in Alto centro	934	361
Punto centrale 3	926	518
Punto in Alto Destra	1275	357
Punto centrale 4	926	518
Punto a Destra	1253	530
Punto centrale 5	930	523
Punto a Sinistra	590	530
Punto centrale 6	924	522
Punto in Basso a Sinistra	600	678
Punto centrale 7	929	525
Punto in Basso Centrale	918	686
Punto centrale 8	919	532
Punto in Basso a Destra	1240	682

3.2.3 Determinazione delle rette di calibrazione

Successivamente, il codice esegue il calcolo delle rette di calibrazione. Per il calcolo delle rette vengono utilizzati:

- i valori mediani di *GazePointX* e *GazePointY*;
- gli angoli di rotazione necessari perché l'occhio fissi il corrispondente punto del diagramma.

Gli angoli reali vengono calcolati utilizzando la formula trigonometrica descritta precedentemente (1). Applicando questa formula alla posizione in cm dei punti della griglia si ottengono gli angoli reali lungo gli assi X e Y.

Tali valori sono riportati nella Tabella 3.3.

Tabella 3.3: Angoli visivi reali associati ai punti del diagramma di calibrazione.

	$\boldsymbol{\theta x}$°	$\boldsymbol{\theta y}$°
Punto a Sinistra	-20°	0°
Punto Centrale	0°	0°
Punto a Destra	+20°	0°
Punto in Basso a Sinistra	-20°	-10°
Punto in Basso Centrale	0°	-10°
Punto in Basso a Destra	+20°	-10°
Punto in Alto a Sinistra	-20°	+10°
Punto in Alto Centrale	0°	+10°
Punto in Alto a Destra	+20°	+10°

I valori mediani e gli angoli reali calcolati vengono quindi utilizzati come input della funzione Matlab *polyfit*, impiegata per eseguire la regressione lineare.
Poiché è necessario ottenere una retta distinta per ciascun asse, vengono eseguite due operazioni separate. All'interno della funzione viene inoltre specificato il valore 1, che indica che il fitting deve essere eseguito tramite una retta.
Nelle figure 3.5 e 3.6 sono riportati i grafici e le equazioni delle rette di calibrazione ottenute.

$$\vartheta_{X\,stimato} = 0.054x - 53.439$$

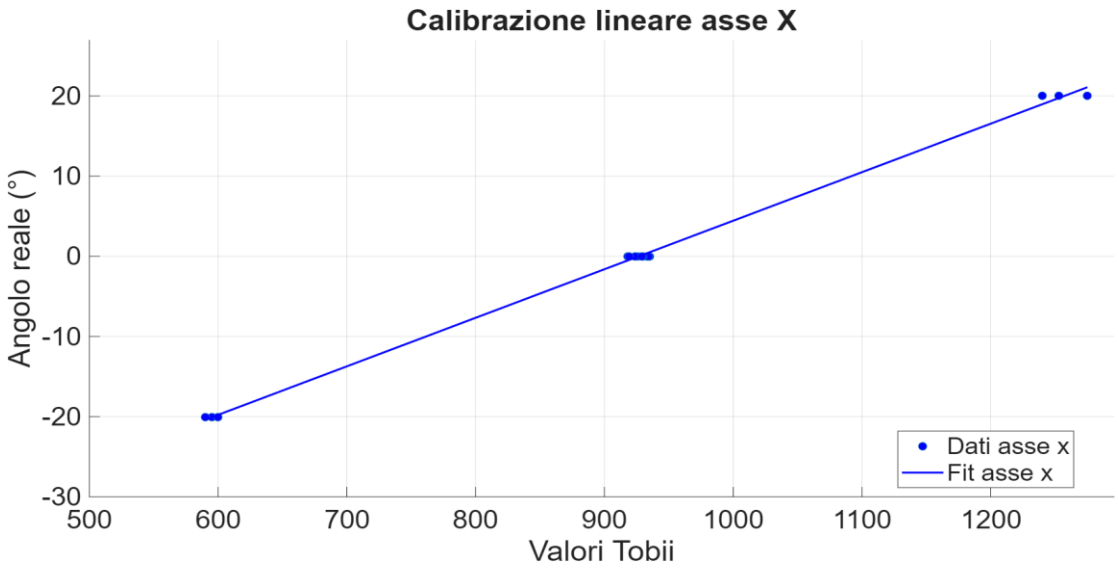

Figura 3.5: Retta di calibrazione per l'asse X.

$$\vartheta_{Y\,stimato} = -0.058y + 21.136$$

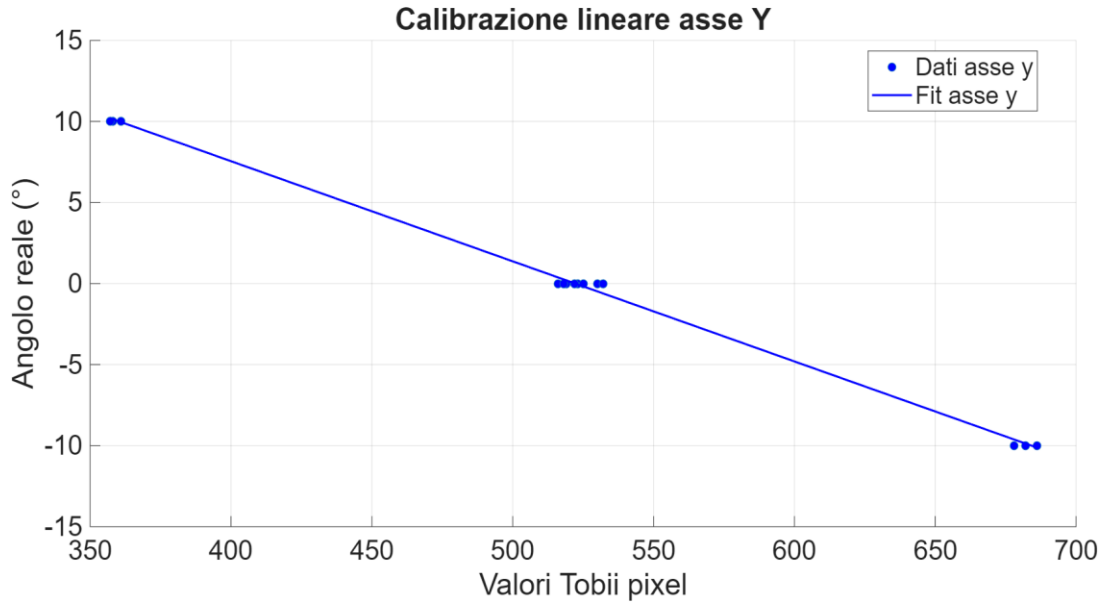

Figura 3.6: Retta di calibrazione per l'asse Y.

In entrambi i grafici è possibile distinguere tre gruppi principali di punti. Come mostrato nella Tabella 3.3, alcuni punti del diagramma di calibrazione condividono lo stesso valore angolare. Di conseguenza, nella calibrazione dell'asse X i punti con lo stesso angolo orizzontale si distribuiscono nello stesso gruppo, mentre nella

calibrazione dell'asse Y si raggruppano i punti con lo stesso angolo verticale. Idealmente, a uno stesso valore angolare dovrebbero corrispondere le stesse coordinate pixel mediane; tuttavia, dai grafici si osserva che i punti appartenenti allo stesso gruppo non coincidono perfettamente. Questo fenomeno è particolarmente evidente nei gruppi centrali, nei quali sono presenti le otto fissazioni del punto centrale del diagramma di calibrazione. Sebbene il punto osservato sia sempre lo stesso, le mediane ottenute risultano leggermente differenti. Questi scostamenti possono essere attribuiti a due fattori: l'errore intrinseco dell'Eye *tracker*, la cui accuratezza dichiarata è pari a circa 0.6° [19], e il fatto che i target disposti sulla griglia abbiano un diametro di circa 2 cm e non siano quindi punti ideali. Di conseguenza, il soggetto può fissare posizioni leggermente diverse all'interno dello stesso target, generando piccole variazioni nelle coordinate registrate dal sistema. Per verificare se le dispersioni osservate fossero compatibili con questi fattori, è stata stimata la dimensione apparente dei target in pixel. A tal fine è stato sviluppato uno script in Python che, a partire da un frame del video di calibrazione, esegue una sogliatura dell'immagine in scala di grigi per segmentare i target dallo sfondo e, successivamente, ne individua i contorni per calcolarne automaticamente il diametro in pixel. I risultati ottenuti sono riportati in Figura 3.7.

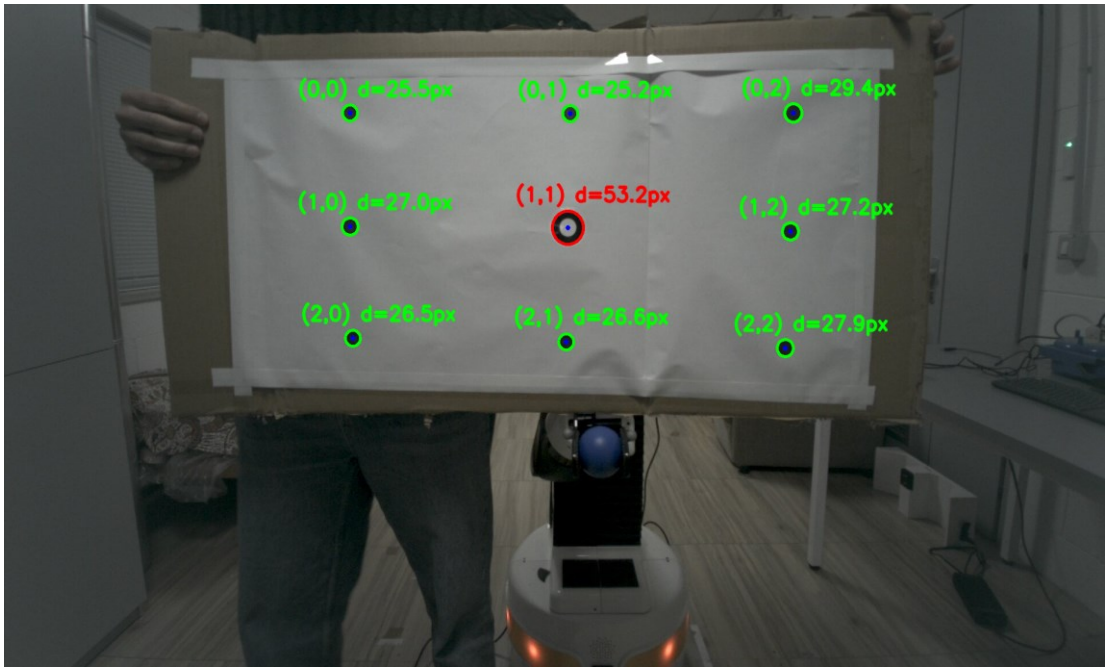

Figura 3.7: Frame del video di calibrazione con i target individuati automaticamente; per ciascun target è riportato il diametro misurato in pixel.

Come mostrato in Figura 3.7, il target centrale del diagramma, utilizzato per la procedura di *one-point calibration,* presenta un diametro di 53 pixel, mentre i target periferici hanno un diametro compreso tra i 27 e i 29 pixel.

Per tenere conto dell'accuratezza dell'Eye Tracker, il valore dell'errore dichiarato dal dispositivo (0.6°) è stato convertito in pixel utilizzando i coefficienti angolari ottenuti dalle rette di calibrazione:

$$e_x = \frac{0.6}{0.054} \approx 11\ pixel$$
$$e_y = \frac{0.6}{0.058} \approx 10\ pixel \quad (2)$$

Poiché i valori ottenuti per i due assi risultano molto simili, è stato adottato il valore maggiore, pari a 11 pixel, come contributo dell'errore strumentale per entrambi gli assi. Sommando tale contributo al diametro dei target misurato in pixel, si ottiene una soglia di tolleranza pari a circa 64 pixel per il target centrale e 40 pixel per i target periferici.

Le soglie di tolleranza ottenute sono state confrontate con la massima differenza osservata tra le mediane dei punti appartenenti allo stesso gruppo angolare.

Per la calibrazione lungo X, il gruppo centrale presenta una dispersione massima di 17 pixel, mentre per la calibrazione Y tale valore è pari a 16 pixel. Entrambi i valori risultano ampiamente inferiori alla soglia di tolleranza stimata, indicando che la dispersione osservata è compatibile con la normale variabilità dovuta sia all'accuratezza dello strumento sia alla possibilità che il soggetto fissi posizioni leggermente differenti all'interno dello stesso target.

Anche per i gruppi periferici gli scostamenti tra le mediane risultano contenuti. Nella calibrazione lungo l'asse X si osserva una differenza massima di 35 pixel nel gruppo a +20° e di 10 pixel nel gruppo a −20°; nella calibrazione lungo l'asse Y, invece, la differenza massima è pari a 8 pixel nel gruppo a −10° e a 3 pixel nel gruppo a +10°.Tutti questi valori risultano inferiori alla soglia di circa 40 pixel.

In conclusione, per tutti i gruppi angolari analizzati, gli scostamenti osservati tra le mediane delle coordinate in pixel risultano compatibili con le soglie di tolleranza stimate.

3.2.4 Valutazione della calibrazione

La parte finale del codice prevede il calcolo di alcune metriche quantitative utilizzate per valutare la bontà della calibrazione ottenuta.

In particolare, sono stati calcolati:

- RMSE (*Root Mean Square Error*);
- errore assoluto (*Absolute Error*, AE);
- coefficiente di determinazione (R^2).

L'*RMSE* rappresenta la radice quadrata dell'errore quadratico medio tra gli angoli stimati dalla retta di calibrazione e gli angoli reali. Valori più bassi indicano una maggiore accuratezza della calibrazione [30]:

$$RMSE_X = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\vartheta_{x_i} - \hat{\vartheta}_{x_i}\right)^2}$$

$$RMSE_Y = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\vartheta_{yi} - \hat{\vartheta}_{y_i}\right)^2}$$

dove:

- N è il numero totale di campioni;
- ϑ_{x_i} e ϑ_{y_i} indicano gli angoli reali;
- $\hat{\vartheta}_{x_i}$ e $\hat{\vartheta}_{y_i}$ rappresentano gli angoli stimati dal modello di calibrazione.

L'errore assoluto rappresenta invece la differenza, in valore assoluto, tra l'angolo reale e quello stimato per ciascun campione acquisito [30]:

$$AE_{x_i} = |\, \vartheta_{x_i} - \hat{\vartheta}_{x_i}|$$

$$AE_{y_i} = |\, \vartheta_{y_i} - \hat{\vartheta}_{y_i}|$$

dove:

- ϑ_{x_i} e ϑ_{y_i} indicano gli angoli reali;
- $\hat{\vartheta}_{x_i}$ e $\hat{\vartheta}_{y_i}$ rappresentano gli angoli stimati dal modello di calibrazione.

Infine, è stato calcolato il coefficiente di determinazione R^2, utilizzato per valutare quanto bene il modello lineare descriva la relazione tra i dati acquisiti e gli angoli reali. Il suo valore è compreso tra 0 e 1, valori prossimi a 1 indicano un'elevata correlazione lineare e quindi una calibrazione più accurata [30]:

$$R^2{}_x = 1 - \frac{\sum_i \left(\vartheta_{x_i} - \hat{\vartheta}_{x_i}\right)^2}{\sum_i (\vartheta_{x_i} - \bar{\vartheta}_x)^2}$$

$$R^2{}_y = 1 - \frac{\sum_i \left(\vartheta_{yi} - \hat{\vartheta}_{y_i}\right)^2}{\sum_i (\vartheta_{y_i} - \bar{\vartheta}_y)^2}$$

- ϑ_{x_i} e ϑ_{y_i} rappresentano i valori reali degli angoli nelle direzioni orizzontale e verticale;
- $\hat{\vartheta}_{x_i}$ e $\hat{\vartheta}_{y_i}$ rappresentano i valori stimati dal modello di calibrazione;
- $\bar{\vartheta}_x$ e $\bar{\vartheta}_y$ indicano i valori medi degli angoli reali;
- $\sum_i (\vartheta_i - \hat{\vartheta}_i)^2$ rappresenta la somma dei quadrati degli errori residui;
- $\sum_i (\vartheta_i - \bar{\vartheta})^2$ rappresenta la variabilità totale dei dati rispetto alla media.

Nella Tabella 3.4 sono riportati i risultati ottenuti per entrambi gli assi.

Tabella 3.4: Metriche quantitative della calibrazione : errore assoluto (AE) calcolato per ciascun punto di fissazione sugli assi X e Y; R^2 e RMSE complessivi per entrambi gli assi.

	Asse X	**Asse Y**
	AE	**AE**
Punto centrale 1	0.382°	0.197°
Punto in Alto Sinistra	0.041°	0.132°
Punto centrale 2	0.503°	0.382°
Punto in Alto centro	0.443°	0.053°
Punto centrale 3	0.042°	0.259°
Punto in Alto Destra	1.048°	0.194°
Punto centrale 4	0.42°	0.259°
Punto a Destra	0.284°	0.482°
Punto centrale 5	0.201°	0.050°
Punto a Sinistra	0.344°	0.482°
Punto centrale 6	0.261°	0.012°
Punto in Basso a Sinistra	0.140°	0.385°
Punto centrale 7	0.526°	0.174°
Punto in Basso Centrale	0.465°	0.109°
Punto centrale 8	0.204°	0.606°
Punto in Alto Sinistra	1.071°	0.138°
$\mathbf{R^2}$	0.998	0.998
RMSE	0.480°	0.298°

Come riportato nella Tabella 3.4, il valore del coefficiente di determinazione R^2 è pari a 0.998 per entrambe le calibrazioni, indicando che il modello lineare descrive in modo accurato la relazione tra i valori angolari del diagramma di calibrazione e le coordinate in pixel registrate dall'*Eye Tracker.*

Anche i valori di RMSE risultano contenuti. In particolare, la calibrazione lungo l'asse Y presenta un errore quadratico medio inferiore rispetto a quella lungo l'asse

X, con valori pari rispettivamente a 0.298° e 0.480°. Entrambi gli errori risultano inferiori a 0.5°, evidenziando una buona accuratezza complessiva del modello di calibrazione.

Analizzando gli errori assoluti dei singoli punti, nella calibrazione lungo l'asse X i valori più elevati sono pari a 1.071° e 1.048°, rispettivamente per i target posti nell'angolo superiore sinistro e nell'angolo superiore destro del diagramma di calibrazione. Nella calibrazione lungo l'asse Y, invece, gli errori assoluti risultano complessivamente inferiori e non raggiungono mai 1°, con un valore massimo pari a 0.606°.

Nel complesso, gli errori osservati nella calibrazione lungo l'asse X risultano lievemente superiori rispetto a quelli ottenuti lungo l'asse Y. Tale comportamento è coerente con la diversa eccentricità dei target impiegati per la calibrazione dei due assi. I punti utilizzati per la calibrazione dell'asse X sono infatti posizionati fino a ±20° dal punto di calibrazione, mentre quelli dell'asse Y raggiungono un'eccentricità massima di ±10°. Poiché nei sistemi basati su *one-point calibration* l'accuratezza diminuisce progressivamente all'aumentare della distanza dal punto di calibrazione, è plausibile che i target più periferici presentino errori leggermente maggiori [29].

Nel complesso, le metriche ottenute evidenziano una buona qualità della calibrazione, confermando che il modello lineare fornisce una rappresentazione accurata della relazione tra valori angolari e coordinate in pixel.

Capitolo 4: Acquisizioni di inseguimento del target

4.1 Tipologie di acquisizioni

Nel corso dello studio sono state effettuate diverse acquisizioni di inseguimento del target presso il Living Lab dell'Università di Pavia. Come descritto in precedenza, durante queste acquisizioni il soggetto deve seguire con lo sguardo un target in movimento a velocità costante; questo tipo di movimento oculare prende il nome di *smooth pursuit*. Per generare il movimento del target è stato utilizzato il robot TIAGo.

Nel presente lavoro sono stati considerati due diversi tipi di movimento del target:

- Movimento lineare: il target segue una traiettoria rettilinea tra due posizioni predefinite. Nella configurazione standard, il robot mantiene il target a 95 cm dal suolo e lo sposta con una velocità media di 0.25 m/s;
- Movimento ad arco: il target segue una traiettoria circolare centrata sulla spalla del paziente, mantenendo una distanza costante da quest'ultima. Nella configurazione standard, il target è posizionato a 95 cm dal suolo e il movimento è parametrizzato mediante *waypoint* consecutivi con un tempo di esecuzione di 1 s per *waypoint.*

Per quanto riguarda i target utilizzati, sono stati scelti oggetti di uso comune e di facile reperibilità: una palla blu, un blocco arancione, uno smartphone con schermo nero, uno smartphone con schermo verde e una borraccia. Per gli oggetti "palla blu" e "blocco arancione" sono stati acquisiti sia i movimenti lineari sia i movimenti ad arco. Per gli altri oggetti è stato invece acquisito esclusivamente il movimento ad arco, poiché durante il movimento lineare la visibilità laterale dell'oggetto risultava ridotta, rendendo più difficile il rilevamento dell'oggetto.

Ai fini dell'organizzazione dei dati raccolti, le acquisizioni sono state suddivise in tre macrocategorie, denominate Acquisizioni di Training, Testing e Gaze. Questa classificazione è stata introdotta esclusivamente in funzione del ruolo svolto da

ciascun gruppo di acquisizioni all'interno del workflow sperimentale. Di conseguenza, al loro interno, le macrocategorie contengono video con caratteristiche differenti. Le tre categorie sono così definite:

- Acquisizioni di Training: destinate alla costruzione dei dataset per l'addestramento delle reti YOLO;
- Acquisizioni di Testing**:** utilizzate per valutare la capacità di generalizzazione delle diverse reti allenate, al fine di individuare l'architettura migliore;
- Acquisizioni di Gaze: dedicate ai risultati finali relativi ai movimenti oculari.

Nella Tabella 4.1 sono sintetizzate le caratteristiche delle diverse acquisizioni.

Tabella 4.1: Tabella riassuntiva delle principali caratteristiche delle Acquisizioni di Training, Testing e Gaze

	Acquisizioni di Training		**Acquisizioni di Testing**		**Acquisizioni di Gaze**
	Tipologia 1	**Video di Sfondo**	**Tipologia 1**	**Tipologia 2**	
Scopo	Addestramento rete YOLO	Addestramento rete YOLO	Test di generalizzazione rete YOLO	Test di generalizzazione rete YOLO	Calcolo angoli
Setup Spaziale	Configurazione 1	Configurazione 2	Configurazione 2	Configurazione 2	Configurazione 2
Setup del Robot	Velocità e altezza standard	Robot fermo, nessun target	Velocita e altezza variate	Velocità e altezza standard	Velocità e altezza standard
Oggetti	Set completo (tutti)	Nessuno	Set completo (tutti)	Set completo (tutti)	Solo palla blu e blocco arancione
Task Soggetto	Seguire il target con lo sguardo	Osservare liberamente l'ambiente	Seguire il target con lo sguardo	Osservare liberamente l'ambiente	Sguardo centrale
Numero video totali	7 video	1 video	28 video	7 video	2 video

Le diverse tipologie di acquisizione si distinguono principalmente per cinque aspetti:

- Il setup spaziale: relativo al *layout* dell'ambiente sperimentale e alla distanza soggetto-target;

- Il task assegnato al soggetto: ovvero il comportamento visivo richiesto al partecipante;
- Il setup del robot: che definisce la velocità di esecuzione del movimento e l'altezza a cui viene mantenuto il target;
- Gli oggetti utilizzati: riguardante la specifica selezione dei target impiegati durante le prove.
- Numero di video: che indica la quantità complessiva di registrazioni effettuate per ogni specifica tipologia.

Per quanto riguarda il setup spaziale, si possono distinguere due tipologie di configurazioni:

- Configurazione 1: il soggetto è seduto su una sedia a una distanza soggetto-target di 1 m e viene utilizzata una configurazione dello sfondo definita di Tipo 1;
- Configurazione 2: il soggetto è seduto sulla medesima sedia, ma la distanza soggetto-target è ridotta a 85 cm e viene utilizzata una configurazione dello sfondo definita di Tipo 2.

4.1.1 Acquisizioni di Training

All'interno delle Acquisizioni di Training sono presenti due differenti tipologie di acquisizioni.

Nella prima tipologia, durante il task, il soggetto segue con lo sguardo il target movimentato dal robot. La velocità del movimento e l'altezza del target vengono mantenute ai valori standard definiti in precedenza, impiegando il set completo di oggetti. Per ciascun oggetto e per ogni traiettoria è stata effettuata una specifica acquisizione, ottenendo complessivamente 7 video.

Nella Figura 4.1 è mostrato un esempio di frame con target palla blu.

Figura 4.1: Esempio di frame estratto dalle Acquisizioni di Training (Prima Tipologia), in cui sono visibili l'oggetto palla blu e la Configurazione spaziale 1.

La seconda tipologia comprende un unico video, definito "video di sfondo", in cui il robot rimane fermo e privo di target, mentre il soggetto osserva liberamente l'ambiente circostante (Figura 4.2). Lo scopo di questa acquisizione è valutare se l'inclusione nei dataset di immagini contenenti esclusivamente lo sfondo possa contribuire a ridurre il numero di falsi positivi generati dalla rete YOLO durante l'analisi delle acquisizioni di Testing, condotte nello stesso set up spaziale.

Figura 4.2: Frame estratto dalle Acquisizioni di Training (Seconda tipologia). Il robot è statico e privo di target all'interno della Configurazione spaziale 2.

4.1.2 Acquisizioni di Testing

Per verificare se i modelli allenati fossero in grado di mantenere una buona precisione anche davanti a variazioni della scena, nelle Acquisizioni di Testing sono state introdotte alcune modifiche rispetto alle acquisizioni di Training. All'interno di questa categoria si distinguono due tipologie di acquisizioni.

Nella prima tipologia, rispetto alle acquisizioni di Training, viene utilizzata la Configurazione 2 e vengono modificati alcuni parametri del robot. In particolare, l'altezza del target viene aumentata e diminuita di 10 cm rispetto al valore standard, mentre la velocità del movimento viene impostata a 1.25× e 1.50× rispetto a quella standard. Applicando queste variazioni ai 7 video di base, sono stati ottenuti complessivamente 28 video.

La Figura 4.3 mostra un esempio di frame estratto dalle acquisizioni di testing della prima tipologia, in cui il target è costituito da una palla blu.

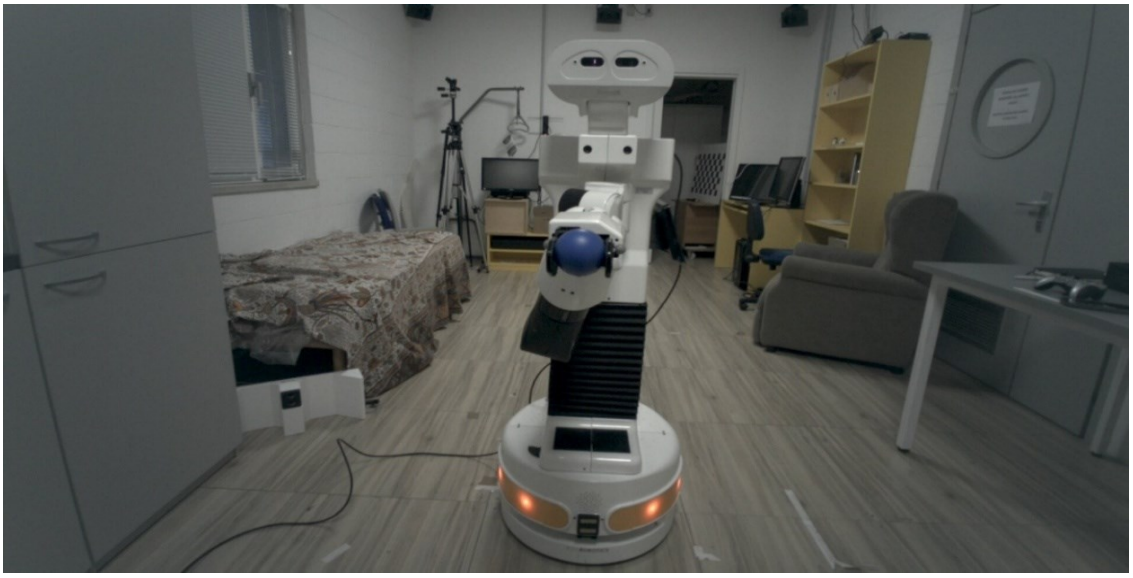

Figura 4.3: Esempio di frame estratto dalle Acquisizioni di Testing (Prima tipologia). È visibile il passaggio alla Configurazione spaziale 2 e la variazione dell'altezza del target a +10 cm.

Nella seconda tipologia, oltre al passaggio alla Configurazione spaziale 2, cambia il compito richiesto al soggetto. Al partecipante viene infatti chiesto di non seguire il target, ma di guardare liberamente l'ambiente circostante. In questo modo il target si trova nelle zone periferiche (ai lati dello schermo) e non al centro del video, permettendo di testare la robustezza della rete in scenari non convenzionali. Questa sessione conta un totale di 7 video.

Un esempio di frame acquisito in queste condizioni è mostrato in Figura 4.4.

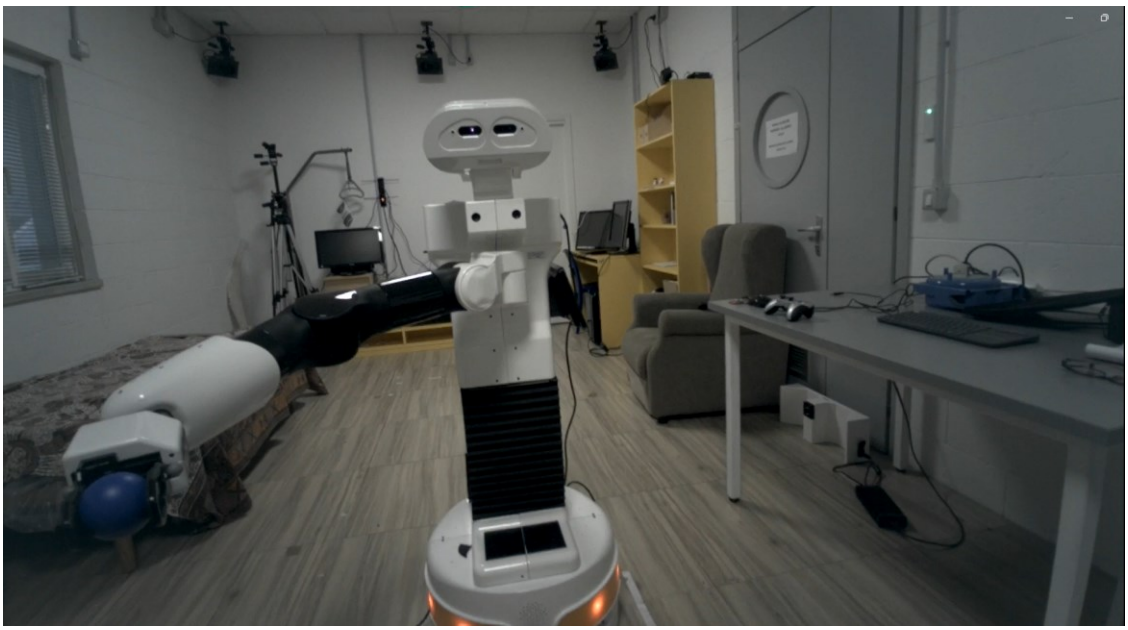

Figura 4.4: Esempio di frame appartenente alle acquisizioni di Testing in Configurazione spaziale 2, in cui il target si trova in una regione periferica del campo visivo.

4.1.3 Acquisizioni di Gaze

Le acquisizioni di Gaze sono state effettuate in continuità con la procedura di calibrazione descritta in precedenza. Su queste registrazioni vengono eseguiti i calcoli finali relativi agli angoli di inseguimento oculare. Per questa categoria viene mantenuta la Configurazione Spaziale 2 e il setup del robot prevede valori di velocità e altezza standard. A differenza delle sessioni precedenti, gli oggetti utilizzati non comprendono l'intero set, ma esclusivamente la palla blu e il blocco arancione, entrambi movimentati lungo una traiettoria ad arco. Durante il task, il soggetto è istruito a seguire il target esclusivamente nella porzione centrale della traiettoria, interrompendo volontariamente l'inseguimento quando l'oggetto raggiunge le regioni più laterali del campo visivo. Questa scelta sperimentale è stata adottata per simulare una condizione assimilabile alla ridotta esplorazione dello spazio tipica dei soggetti affetti da eminattenzione spaziale [3].

L'elaborazione di queste acquisizioni si articola in due fasi principali. Nella prima fase vengono costruiti diversi dataset mediante la piattaforma Roboflow utilizzando le acquisizioni di Training. I dataset ottenuti vengono successivamente impiegati per addestrare e valutare differenti modelli YOLO in ambiente Python. Le prestazioni dei modelli vengono dapprima analizzate sui rispettivi Test Set e successivamente confrontate sulle acquisizioni di Testing con l'obiettivo di valutarne la capacità di generalizzazione e selezionare il modello più performante. In questa fase vengono utilizzati esclusivamente i video acquisiti dalla telecamera di scena privi della sovrapposizione del gaze.

Nella seconda fase viene utilizzato un ulteriore script Python che riceve come input il modello YOLO selezionato, i video delle acquisizioni di Gaze e le rette di calibrazione ottenute nella fase precedente. L'obiettivo è determinare, per ogni frame, la posizione del target e quella dello sguardo del soggetto. Le coordinate ottenute, inizialmente espresse in pixel, vengono successivamente convertite in valori angolari. Per questa elaborazione sono necessari sia i video della scena senza gaze overlay, sia i file Excel contenenti i dati di gaze esportati da Tobii Pro Lab.

4.2 Roboflow e costruzione dei dataset

La prima fase del lavoro consiste nella costruzione dei dataset utilizzati per l'addestramento delle reti YOLO a partire dalle Acquisizioni di Training. In questo contesto, un dataset è composto da una raccolta di immagini (estratte dai video) associate alle relative annotazioni. L'annotazione è il processo attraverso il quale si forniscono alla rete neurale le informazioni necessarie per riconoscere e localizzare i target nella scena. Nello specifico, per ogni immagine vengono definiti due elementi:

- La classe: l'etichetta che dice alla rete qual è l'oggetto presente nell'immagine (ad esempio "palla blu" o "borraccia");
- La *bounding box*: un riquadro rettangolare disegnato attorno all'oggetto per indicarne l'esatta posizione nella scena.

Queste informazioni consentono al modello YOLO di imparare a riconoscere e localizzare gli oggetti nella scena durante la fase di addestramento [31].

Per la creazione dei dataset è stata utilizzata la piattaforma Roboflow [32], un ambiente web dedicato alla preparazione dei dataset per applicazioni di computer vision. La piattaforma consente di gestire tutte le fasi precedenti all'addestramento del modello, tra cui l'estrazione dei frame dai video, l'annotazione delle immagini, la configurazione del dataset e la sua esportazione nel formato compatibile con YOLO. Attraverso Roboflow sono stati costruiti tre dataset differenti: il primo comprende esclusivamente i frame estratti dai video della prima tipologia, mentre negli altri due sono stati aggiunti anche i frame del "video di sfondo".

4.2.1 Estrazione dei frame

Una volta importata ciascuna acquisizione su Roboflow, si definisce il numero di immagini da estrarre. È importante che le immagini selezionate siano il più possibile differenti tra loro, poiché un dataset troppo ridondante può ridurre la capacità di generalizzazione della rete [33]. Nel caso specifico, i video della prima tipologia hanno una durata di circa 30 secondi e presentano condizioni di acquisizione sostanzialmente stabili. Di conseguenza, i fotogrammi consecutivi

risultano molto simili tra loro. Per evitare di introdurre immagini ridondanti nel dataset, si è scelto di estrarre, per ciascun video, circa 50 frame ritenuti sufficientemente differenti da garantire una buona variabilità delle immagini. Per quanto riguarda invece il video di sfondo, in letteratura è consigliato che le immagini prive di target non superino circa il 10% del numero totale di immagini del dataset [34]. Poiché il dataset privo di immagini di sfondo risultava composto da circa 330 immagini, il numero massimo consigliato di frame di sfondo è circa 30. In particolare, per il secondo dataset sono stati selezionati 29 frame di sfondo e nel terzo ne sono stati aggiunti 11.

4.2.2 Annotazione dei frame estratti

Una volta completata l'estrazione dei frame, si passa alla fase di annotazione delle immagini. In particolare, per ciascun target viene disegnata una *bounding box*, ossia un riquadro rettangolare che ne delimita la posizione all'interno della scena. Per garantire un addestramento efficace, la *bounding box* deve aderire il più possibile ai contorni visibili dell'oggetto, fornendo alla rete informazioni precise sulla sua localizzazione e sulle sue dimensioni [31].

L'annotazione può essere effettuata manualmente oppure tramite la funzione *Auto Label* di Roboflow, che genera automaticamente la *bounding box* a partire da brevi descrizioni testuali fornite dall'utente[35].

Le due modalità di annotazione sono illustrate in Figura 4.6.

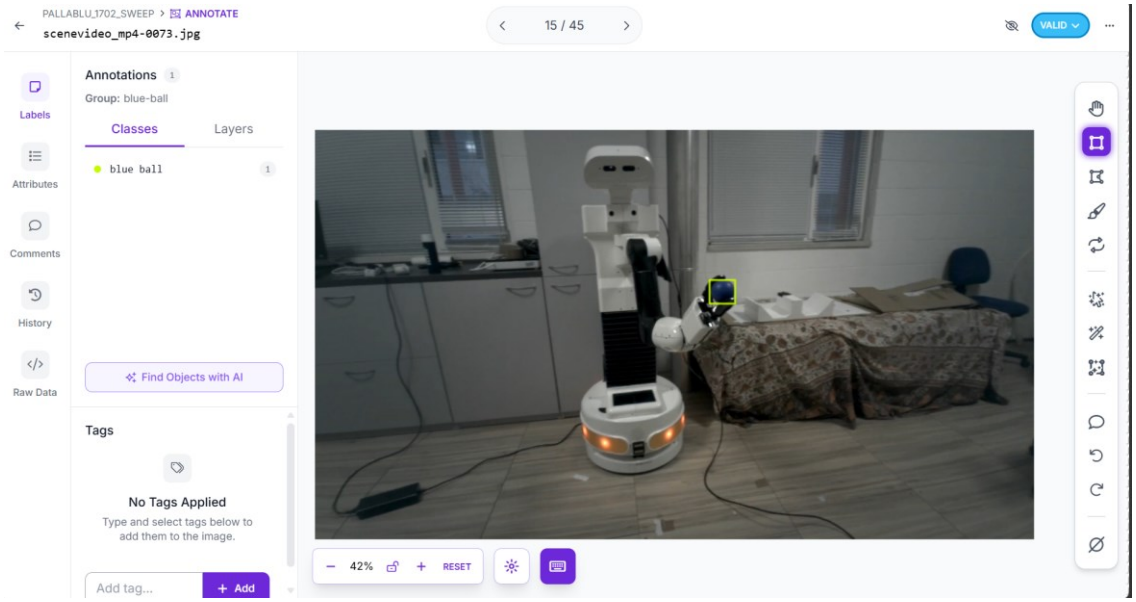

Figura 4.5: Interfaccia di Roboflow durante la fase di annotazione manuale del target mediante bounding box.

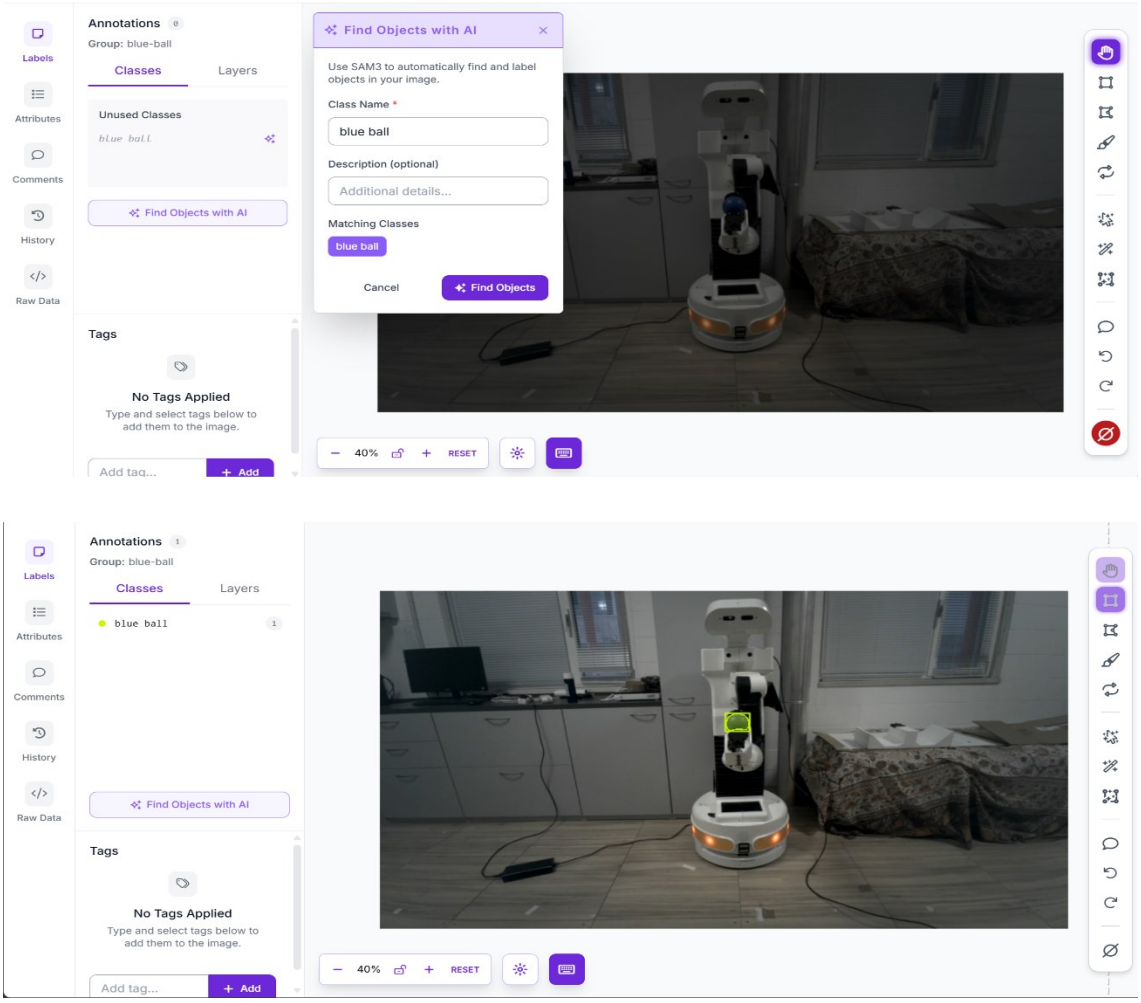

Figura 4.6: Annotazione automatica della palla blu tramite la funzione AutoLabel di Roboflow. La bounding box viene generata automaticamente a partire dall'identificazione dell'oggetto.

Per i frame estratti dal video di sfondo non viene effettuata alcuna annotazione, poiché non sono presenti oggetti di interesse. Terminata questa fase, le immagini provenienti dalle diverse registrazioni vengono unite per costruire i tre dataset finali. Nella Figura 4.7 vengono mostrati i tre dataset realizzati, il primo dataset è composto da 331 immagini, il secondo da 360 immagini e il terzo da 342 immagini.

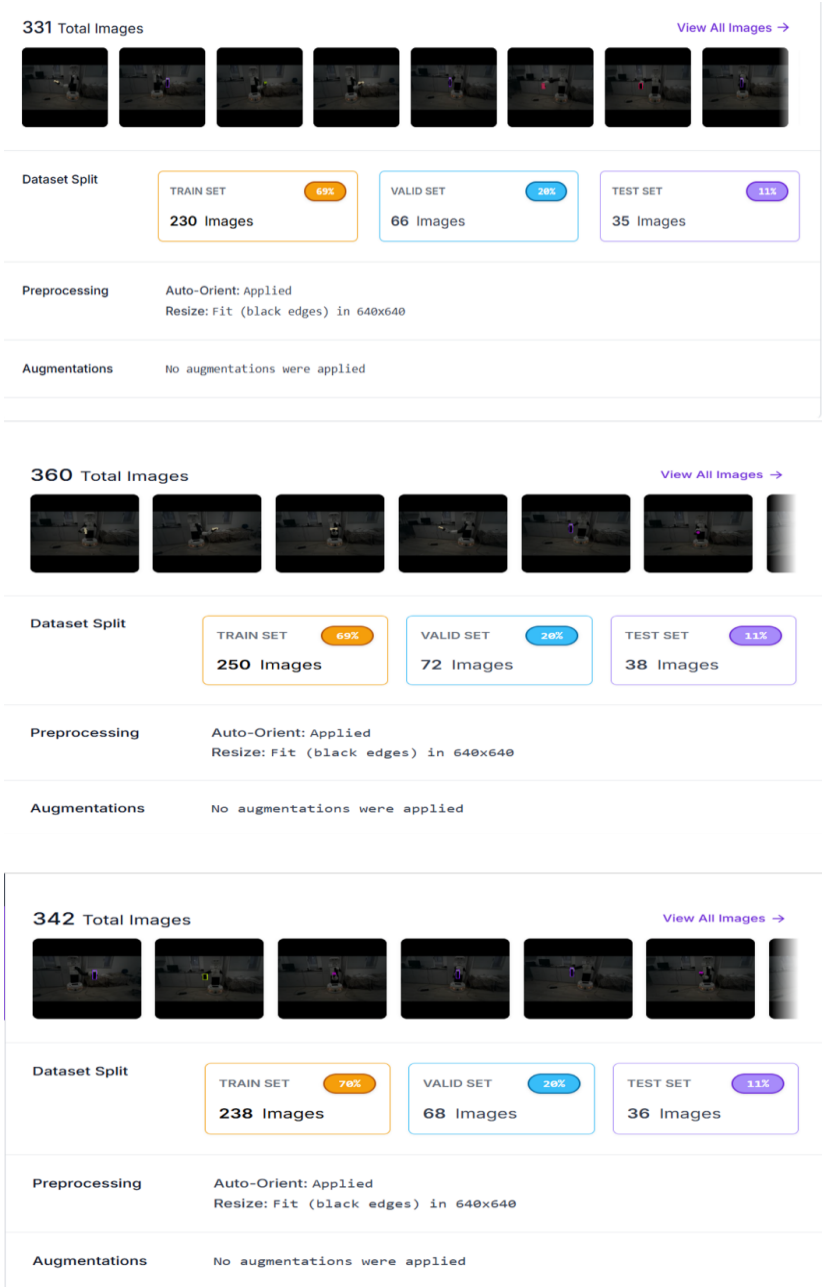

Figura 4.7: Configurazione dei tre dataset utilizzati per l'addestramento dei modelli *YOLO, comprensiva del numero totale di immagini, della suddivisione in train, validation e test e delle operazioni di preprocessing applicate.*

4.2.3 Configurazione ed esportazione dei dataset

Come mostrato nella Figura 4.7, una volta completata la costruzione dei dataset sono stati definiti i parametri di esportazione mantenendo le stesse impostazioni per tutte le configurazioni. Come prima cosa è stato definito il *Dataset Split*, attraverso il quale ciascun dataset è stato suddiviso nei sottoinsiemi di *training*, *validation* e *test.* Per farlo, sono state mantenute le percentuali standard suggerite dalla piattaforma Roboflow, pari approssimativamente al 70%, 20% e 10%, rispettivamente.

Le lievi differenze percentuali visibili nella Figura 4.7 sono dovute all'arrotondamento automatico necessario per assegnare un numero intero di immagini a ciascun sottoinsieme [36], [37].

Per il *preprocessing* è stata mantenuta attiva l'opzione predefinita *Auto-Orient*, che consente di preservare il corretto orientamento delle immagini [37]. Successivamente, tutte le immagini sono state ridimensionate a 640×640 pixel, una risoluzione comunemente utilizzata nell'addestramento dei modelli YOLO poiché consente di ridurre il carico computazionale mantenendo prestazioni adeguate [38]. Poiché le immagini originali presentano un formato rettangolare, un ridimensionamento diretto al formato quadrato richiesto dal modello ne avrebbe alterato le proporzioni. Con l'opzione *Fit (Black Edges)*, invece, l'immagine viene riscalata mantenendo il rapporto d'aspetto originale e gli spazi rimanenti vengono riempiti automaticamente con bordi neri (*padding*) [37].

Infine, in questa fase si è scelto di non applicare tecniche di *Data Augmentation* tramite Roboflow, ma di gestirle successivamente durante la fase di *training* della rete.

Una volta definite le impostazioni di esportazione, ciascun dataset viene scaricato in locale nel formato compatibile con YOLOv8, l'architettura scelta per la successiva fase di addestramento.

L'esportazione genera una cartella principale contenente il file di configurazione *data.yaml* e le tre sottocartelle *train, valid* e *test,* corrispondenti ai sottoinsiemi definiti durante la fase di suddivisione del dataset. Il file data.yaml verrà fornito in input all'architettura YOLO, contiene infatti le informazioni necessarie affinché YOLO possa localizzare correttamente il dataset e interpretarne il contenuto durante la fase di addestramento [39].

All'interno di ciascuna delle tre sottocartelle sono presenti le immagini e i relativi file di annotazione. Le annotazioni contengono le informazioni necessarie per descrivere la classe dell'oggetto e la sua posizione all'interno dell'immagine. Il formato di questi file dipende dalla modalità di annotazione utilizzata in precedenza in Roboflow.

Nel caso delle annotazioni manuali, ciascun file contiene cinque valori: il primo rappresenta l'identificatore numerico associato alla classe dell'oggetto, mentre i quattro valori successivi descrivono la *bounding box* tramite le coordinate del centro (x, y), la larghezza e l'altezza. Tutte le coordinate sono normalizzate rispetto alle dimensioni dell'immagine e assumono quindi valori compresi tra 0 e 1 [40].

Nel caso delle annotazioni generate tramite la funzione *Auto Label*, invece, l'oggetto viene descritto tramite una sequenza di punti che ne definiscono il contorno, producendo una annotazione di segmentazione. In questo caso, il file contiene l'identificatore numerico della classe seguito dalle coordinate normalizzate dei vertici del poligono che delimita l'oggetto [39].

Questa differenza non rappresenta un problema per la successiva fase di addestramento, poiché YOLOv8 include la funzione *segment2box*, che converte automaticamente le annotazioni di segmentazione nella corrispondente *bounding box* [41].

4.3 Addestramento delle reti YOLO

La fase successiva del lavoro riguarda l'addestramento delle reti YOLO (*You Only Look Once*) [42] sui dataset generati nella fase precedente. YOLO è una famiglia di reti neurali progettata per applicazioni di *object detection*, in grado di identificare automaticamente gli oggetti presenti in un'immagine, determinarne la classe di appartenenza e localizzarne la posizione mediante *bounding box*. Nel presente lavoro è stata utilizzata l'architettura YOLOv8n [43], dove la lettera *n* indica la versione *nano* del modello. Questa versione è caratterizzata da dimensioni ridotte e da una bassa complessità computazionale, consentendo tempi di addestramento e inferenza contenuti pur mantenendo buone prestazioni in termini di accuratezza. I modelli YOLOv8 vengono inizialmente forniti come reti pre-addestrate su grandi dataset generici, come il dataset COCO, costituito da circa 118.000 immagini di addestramento e contenente 80 classi di oggetti comuni. Nel presente studio, la rete è stata ulteriormente addestrata sui dataset costruiti tramite Roboflow adottando la strategia di *transfer learning*. Questo approccio consente di partire da un modello già pre-addestrato su un ampio dataset generico e di adattarlo a uno specifico compito applicativo, sfruttando le conoscenze acquisite durante il pre-addestramento e specializzandolo nel riconoscimento degli oggetti di interesse del presente studio [44].

L'addestramento dei modelli viene effettuato in ambiente Python, versione 3.12, utilizzando il *framework Ultralytics* YOLO [45], attraverso il quale è stata inizializzata l'architettura YOLOv8n.

Come descritto nella fase di costruzione dei dataset, ciascun dataset esportato da Roboflow è suddiviso in tre sottoinsiemi: *Training Set, Validation Set e Test Set*, corrispondenti alle cartelle *train*, *valid* e *test*. Durante la fase di *training* il modello analizza ripetutamente le immagini del *Training Set*; ogni analisi completa dell'intero dataset corrisponde a un'epoca.

Al termine di ogni epoca viene eseguita una fase di validazione sul *Validation Set* per monitorare le prestazioni del modello durante l'apprendimento. Una volta completato il *training*, il modello viene infine valutato sul *Test Set*, costituito da immagini mai utilizzate nelle fasi precedenti.

4.3.1 Fase di Training e Validation

La fase di *training* e quella di *validation* vengono eseguite congiuntamente mediante la funzione *model.train()*. All'interno della funzione vengono definiti i seguenti parametri: il percorso al file *data.yaml*, il numero di epoche, la dimensione del *batch*, la dimensione delle immagini in input e le tecniche di *Data Augmentation* da applicare.

Il file *data.yaml*, generato durante l'esportazione del dataset da Roboflow, contiene i percorsi alle cartelle *train*, *valid* e *test* e le informazioni relative alle classi presenti nel dataset. Attraverso questo file YOLO è quindi in grado di individuare automaticamente le immagini da utilizzare nelle diverse fasi di addestramento e valutazione.

Nel presente lavoro il numero di epoche è stato impostato a 150, mentre la dimensione del *batch* è stata fissata a 8. Ciò significa che l'intero *Training Set* viene analizzato 150 volte durante l'addestramento e che i parametri della rete vengono aggiornati dopo l'elaborazione di gruppi di 8 immagini alla volta [46].

La dimensione delle immagini in input è stata impostata a 640×640 pixel, valore già definito durante la preparazione dei dataset su Roboflow.

Data augmentation

Durante il *training* vengono inoltre applicate diverse tecniche di *Data Augmentation*, ovvero trasformazioni artificiali applicate alle immagini del *Training Set* per generarne versioni leggermente modificate. Queste tecniche consentono di aumentare la variabilità delle immagini disponibili senza dover acquisire nuovi dati e hanno l'obiettivo di migliorare la capacità di generalizzazione della rete, riducendo il rischio di *overfitting*. Le trasformazioni vengono applicate dinamicamente (*on-the-fly*): le immagini aumentate non vengono generate e salvate preventivamente sul disco, ma create direttamente durante l'addestramento. In questo modo, a ogni epoca la rete può osservare versioni differenti delle stesse immagini, aumentando ulteriormente la variabilità del dataset. Di fatto, con questo approccio la rete viene allenata su un numero di immagini pari al prodotto tra numero di immagini nel dataset di training e il numero di epoche [47], [48].

Nel presente lavoro sono state definite due differenti configurazioni di *Data Augmentation*; le due configurazioni, illustrate in Figura 4.8, prevedono le stesse trasformazioni, ma differiscono per i valori assegnati ad alcuni parametri.

```
degrees=30.0,
translate=0.3,
shear=10.0,
scale=0.6,
fliplr=0.5,

hsv_h=0.015,
hsv_s=0.6,
hsv_v=0.6,

mosaic=1.0,
mixup=0.4,
```

```
degrees=20.0,
translate=0.15,
shear=5.0,
scale=0.4,
fliplr=0.5,

hsv_h=0.015,
hsv_s=0.5,
hsv_v=0.4,

mosaic=1.0,
mixup=0.2,
```

Figura 4.8: Configurazioni di Data Augmentation.

La configurazione mostrata a sinistra, definita in questo lavoro come Data Augmentation di tipo 1, utilizza valori più elevati per alcuni parametri e genera pertanto trasformazioni più marcate delle immagini.

La configurazione riportata a destra, definita come Data Augmentation di tipo 2, utilizza invece valori più contenuti, producendo modifiche meno aggressive.

I parametri *fliplr*, *hsv_h* e *mosaic* sono stati mantenuti ai valori standard previsti da YOLO in entrambe le configurazioni. Per gli altri parametri sono invece stati definiti valori differenti, al fine di valutare l'effetto di livelli diversi di *Data Augmentation* sulle prestazioni del modello.

Le trasformazioni utilizzate possono essere suddivise in tre categorie principali:

- trasformazioni geometriche;
- trasformazioni di illuminazione;
- tecniche di composizione.

Tra le trasformazioni geometriche sono state utilizzate la rotazione (*degrees*), la deformazione prospettica (*shear*), la traslazione (*translate*), il ridimensionamento (*scale*) e il ribaltamento orizzontale (*fliplr*). Queste operazioni permettono di simulare variazioni nella posizione, nell'orientamento e nelle dimensioni apparenti degli oggetti presenti all'interno della scena.

Per simulare differenti condizioni di illuminazione sono state introdotte variazioni sui canali dello spazio colore HSV. In particolare, *hsv_h* modifica la tonalità dell'immagine, *hsv_s* la saturazione e *hsv_v* la luminosità.

Infine, sono state utilizzate due tecniche di composizione:

- *Mosaic*: combina quattro immagini differenti in un’unica immagine,
- *MixUp*: che sovrappone parzialmente due immagini e le relative annotazioni.

Queste tecniche aumentano la variabilità del dataset e aiutano la rete a riconoscere gli oggetti anche in presenza di occlusioni o sfondi complessi [47].

Risultati dell'addestramento

Al termine del *training*, il *framework* genera automaticamente una cartella contente i risultati dell'addestramento. In particolare, vengono salvati i pesi della rete, diversi grafici relativi alle prestazioni del modello e alcuni risultati qualitativi utili per analizzarne il comportamento.

Per quanto riguarda i pesi, vengono prodotti due file principali: *last.pt,* che contiene i pesi ottenuti all'ultima epoca di addestramento, e *best.pt,* che contiene invece i pesi che hanno ottenuto le migliori prestazioni sul *Validation Set*. Quest'ultimo file verrà successivamente utilizzato per eseguire l'inferenza sui nuovi video.

I risultati grafici esportati comprendono una figura contenente l'andamento delle principali *loss* e metriche durante le diverse epoche di addestramento, le curve *Precision-Recall, Precision-Confidence, Recall-Confidence e F1-Confidence,* che descrivono il comportamento del modello al variare della soglia di confidenza, e le matrici di confusione.

La prima figura è costituita da dieci grafici. I primi sei rappresentano le *Loss Curves*, ovvero l'andamento dell'errore sul *Training Set* e sul *Validation Set*. In particolare, la *Box Loss* misura l'errore nella localizzazione delle *bounding box*, la *Cls Loss* l'errore nella classificazione degli oggetti e la *DFL Loss* (*Distribution Focal Loss*) la precisione nella definizione dei bordi delle *bounding box*. I restanti quattro grafici mostrano l'andamento delle principali metriche di valutazione calcolate sul *Validation Set*. La *Precision* misura la percentuale di rilevamenti corretti tra tutti quelli effettuati dal modello, mentre la *Recall* quantifica la capacità di individuare gli oggetti realmente presenti nelle immagini. La *mAP50* (*mean Average Precision* con soglia IoU pari al 50%) valuta la capacità del modello di riconoscere e localizzare correttamente gli oggetti considerando valida una predizione quando la *bounding box* predetta si sovrappone a quella reale per

almeno il 50%. La *mAP50-95* fornisce invece una valutazione più rigorosa delle prestazioni, poiché considera soglie IoU comprese tra il 50% e il 95%.

Le curve relative alla soglia di confidenza comprendono la curva *Precision-Recall*, che descrive il compromesso tra precisione e *recall* del modello, la curva *Precision-Confidence*, che mostra come varia la precisione al variare della soglia di confidenza, la curva *Recall-Confidence*, che descrive l'andamento del *recall* al variare della stessa soglia, e la curva *F1-Confidence*, che permette di individuare il valore di confidenza che massimizza l'equilibrio tra precisione e *recall*.

Vengono inoltre esportate le matrici di confusione calcolate sul *Validation Set*, che consentono di confrontare le classi reali degli oggetti con quelle predette dal modello. Esse possono essere espresse in forma assoluta, riportando il numero di campioni classificati in ciascuna categoria, oppure in forma normalizzata, riportando valori relativi. In una matrice di confusione normalizzata ideale, tutti i valori sono pari a 1 lungo la diagonale principale e a 0 nelle altre celle, indicando una classificazione perfetta.

Tra i risultati qualitativi vengono infine salvate immagini dei *batch* di addestramento, utili per osservare gli effetti delle tecniche di *Data Augmentation* applicate durante il *training*, e coppie di immagini del *Validation Set* nelle quali sono riportate, affiancate, le annotazioni di riferimento (*ground truth*) realizzate durante la costruzione del dataset in Roboflow e le corrispondenti *bounding box* predette dalla rete [49].

4.3.2 Valutazione sul Test Set

Terminata la fase di *training*, ciascun modello viene valutato sul *Test Set*, costituito da immagini mai utilizzate durante l'addestramento. Per questa analisi vengono caricati i pesi contenuti nel file *best.pt* e la valutazione viene eseguita tramite la funzione *model.val()* specificando il percorso del file *data.yaml* e impostando il parametro *split='test'*. Anche sul *Test Set* vengono calcolate le stesse metriche utilizzate nella fase di addestramento. Tuttavia, poiché il *Test Set* viene valutato

una sola volta mediante il modello già addestrato, non vengono generate curve temporali delle prestazioni, ma un unico valore finale per ciascuna metrica, rappresentativo delle prestazioni del modello sul *Test Set* [50].

4.4 Configurazioni dei modelli addestrati

A partire dai tre dataset costruiti tramite Roboflow sono stati addestrati diversi modelli YOLO utilizzando le due configurazioni di *Data Augmentation* descritte in precedenza.

In particolare, il Dataset 1 e il Dataset 2 sono stati addestrati sia con la configurazione di Data Augmentation di tipo 1 sia con la Data Augmentation di tipo 2. Il Dataset 3 invece, poiché è introdotto in una fase successiva come verifica aggiuntiva, è stato addestrato esclusivamente utilizzando la Data Augmentation di tipo 2.

Combinando i diversi dataset con le configurazioni di Data Augmentation sono stati quindi ottenuti cinque differenti modelli YOLO, riassunti nella Tabella 4.2.

Tabella 4.2: Configurazione dei modelli addestrati.

	Dataset Roboflow	**Data Augmentation**
Modello 1	Dataset 1	Data Augmentation 1
Modello 2	Dataset 1	Data Augmentation 2
Modello 3	Dataset 2	Data Augmentation 1
Modello 4	Dataset 2	Data Augmentation 2
Modello 5	Dataset 3	Data Augmentation 2

4.5 Confronto fra i modelli e selezione del modello finale

4.5.1 Valutazione sul Test Set

Dopo aver completato l'addestramento dei diversi modelli, sono state confrontate le prestazioni ottenute durante le fasi di *validation* e di *test*.

Per quanto riguarda la fase di *validation*, tutti i modelli hanno mostrato un andamento molto simile delle *loss* e delle principali metriche di valutazione. Anche le matrici di confusione normalizzate calcolate sul *Validation Set* presentano valori pari a 1 lungo la diagonale principale, indicando una corretta classificazione di tutti i campioni utilizzati per la validazione.

Risultati analoghi sono stati osservati anche sul *Test Set*. Le matrici di confusione normalizzate mostrano infatti valori pari a 1 lungo la diagonale principale per tutti i modelli, mentre le metriche calcolate, riportate nella Tabella 4.3, risultano molto simili tra loro.

Tabella 4.3: Metriche calcolate sul Test Set nei diversi modelli

	Modello 1	Modello 2	Modello 3	Modello 4	Modello 5
mAP50	0.995	0.995	0.995	0.995	0.995
mAP50-95	0.8551	0.8427	0.8462	0.8312	0.8415
Precision	0.8937	0.9756	0.98	0.9771	0.9841
Recall	1	1	1	1	1

Nel complesso, l'analisi dei risultati ottenuti sui rispettivi set di *validation* e *test* non ha evidenziato differenze significative tra i diversi modelli, rendendo difficile individuare una configurazione chiaramente migliore delle altre. Si è pertanto proceduto a un ulteriore confronto tra i modelli utilizzando le acquisizioni di Testing.

4.5.2 Valutazione delle capacità di generalizzazione

Come descritto in precedenza, le acquisizioni di Testing presentano caratteristiche differenti rispetto alle acquisizioni utilizzate per la costruzione dei dataset di training. Per questo motivo, esse rappresentano uno strumento utile per valutare la capacità di generalizzazione dei diversi modelli su dati non impiegati durante la fase di addestramento.

Per eseguire l'inferenza sui nuovi video, viene caricato il file *best.pt* del modello selezionato e, utilizzando il *framework Ultralytics YOLO* [45], il codice analizza i video frame per frame al fine di rilevare il target di interesse. In questa fase viene inoltre definita una soglia di confidenza, ovvero il valore minimo richiesto affinché una predizione venga considerata valida. Se la confidenza associata a una *detection* risulta inferiore a tale soglia, la corrispondente *bounding box* viene scartata. Nel presente lavoro è stato utilizzato un valore di confidenza pari a 0.25, corrispondente all'impostazione standard di YOLO [51].

Nel codice è stato inoltre introdotto un vincolo: nel caso in cui la rete rilevi più oggetti all'interno dello stesso frame, viene mantenuta esclusivamente la predizione caratterizzata dal valore di confidenza più elevato. Tale scelta è stata adottata poiché nei video acquisiti è presente un solo oggetto di interesse per volta, quello tenuto in mano dal robot. Di conseguenza, conservare unicamente la *detection* più affidabile consente di eliminare rilevamenti multipli non necessari e di ridurre possibili falsi positivi dovuti agli oggetti presenti sullo sfondo.

Per ogni frame analizzato vengono salvate in un file Excel le principali informazioni associate alla *detection* selezionata (Figura 4.4).

In particolare, vengono registrati:

- la classe dell'oggetto rilevato;
- il valore di confidenza della predizione;
- le coordinate di due vertici opposti della *bounding box*;

- le coordinate del centroide della *bounding box*.

Tutte le coordinate sono espresse in pixel rispetto all'origine del frame video, posta nell'angolo superiore sinistro dell'immagine [51].

Tabella 4.4: Esempio delle prime righe del file Excel generato.

FRAME	CLASSE	CONFIDENZA	X1_MIN	Y1_MIN	X2_MAX	Y2_MAX	CENTROIDE_X	CENTROIDE_Y
1	blue ball	0,9184	957	536	1045	632	1001	584
2	blue ball	0,9176	956	535	1044	632	1000	584
3	blue ball	0,9267	955	537	1043	631	999	584
4	blue ball	0,9268	955	536	1042	629	999	582
5	blue ball	0,9342	954	535	1042	628	998	582
6	blue ball	0,9287	955	535	1042	626	998	580
7	blue ball	0,9335	954	535	1042	627	998	581
8	blue ball	0,9334	955	535	1043	627	999	581
9	blue ball	0,9347	955	535	1043	628	999	581
10	blue ball	0,9336	955	535	1043	628	999	581

Oltre al file Excel, il codice produce un video di output in cui, per ogni frame, sono sovrapposti all'immagine originale la *bounding box* dell'oggetto rilevato, il suo centroide e il corrispondente valore di confidenza (Figura 4.9) [51].

Figura 4.9: Frame di output della rete YOLO con visualizzazione della bounding box associata al target rilevato.

4.5.3 Confronto dei modelli e selezione del modello finale

Una volta eseguita questa procedura per tutti i video delle acquisizioni di Testing e per tutti i modelli addestrati, sono state costruite le matrici di confusione normalizzate per confrontarne le prestazioni. Nelle matrici le righe rappresentano le classi reali (*ground truth*), mentre le colonne rappresentano le classi predette dalla rete.

Le matrici di confusione normalizzate sono illustrate nelle Figure 4.10 e 4.11.

Figura 4.10: Matrici di confusione normalizzate relative al Modello1 e al Modello 2.

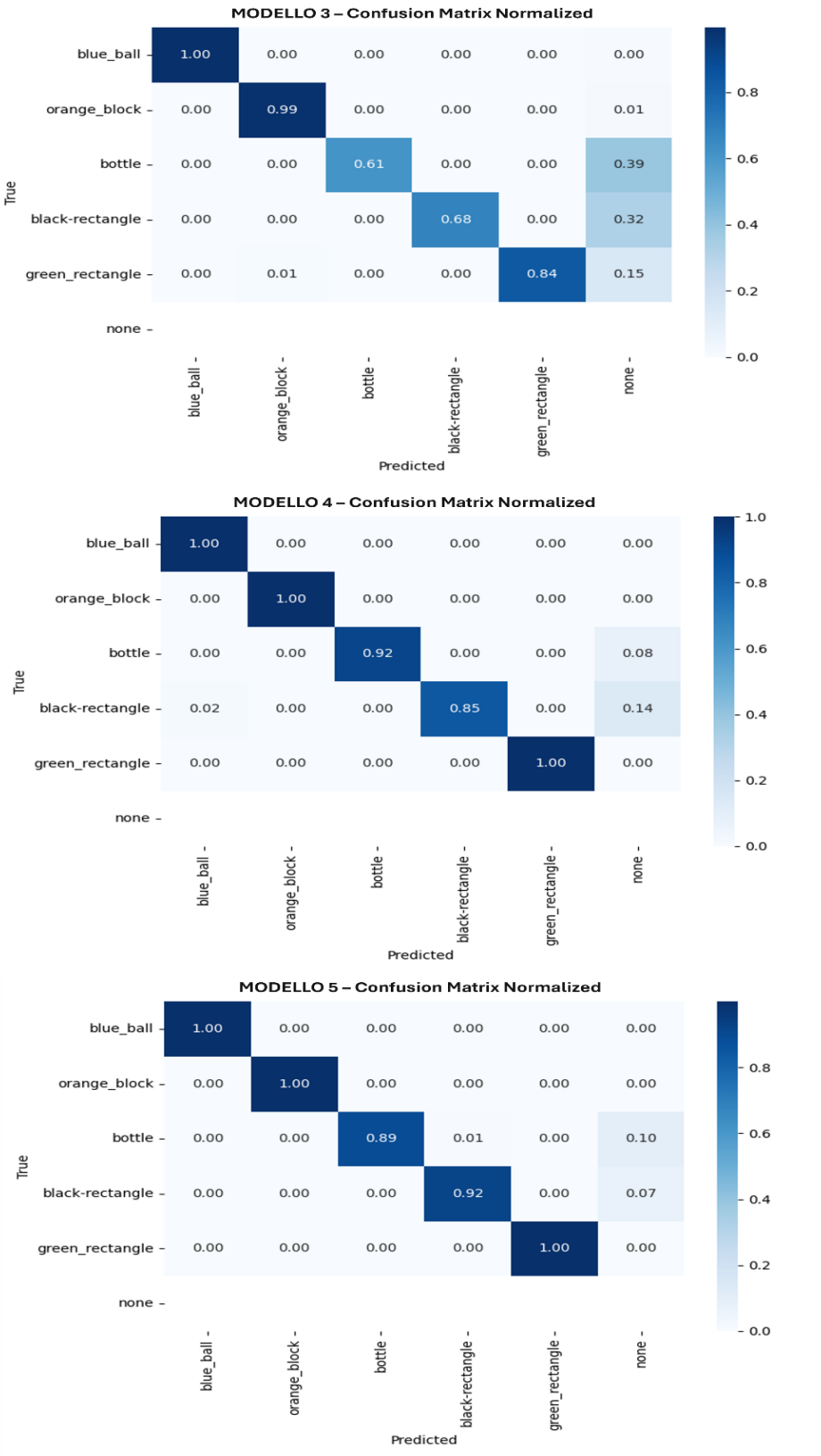

Figura 4.11: Matrici di confusione normalizzate riferite al Modello 3, Modello 4 e Modello 5.

Per valutare l'influenza della configurazione delle *Data Augmentation* sulle prestazioni dei modelli, sono stati confrontati i modelli addestrati sullo stesso dataset utilizzando le due differenti configurazioni di *augmentation*. In particolare, sono stati confrontati il Modello 1 con il Modello 2 e il Modello 3 con il Modello 4. In entrambi i casi, i modelli addestrati con la Data Augmentation di tipo 2 mostrano prestazioni migliori rispetto a quelli addestrati con la Data Augmentation di tipo 1. Il confronto tra Modello 1 e Modello 2 evidenzia infatti un miglioramento per le classi *bottle*, *black-rectangle* e *green_rectangle*, mentre il Modello 4 presenta risultati superiori rispetto al Modello 3 per tutte le classi considerate. Questi risultati suggeriscono che la Data Augmentation di tipo 2, caratterizzata da trasformazioni meno aggressive, sia maggiormente adatta al problema affrontato.

È stato inoltre valutato l'effetto dell'introduzione di frame di sfondo all'interno dei dataset di addestramento. A tal fine sono stati confrontati i modelli addestrati con la stessa configurazione di *Data Augmentation* ma con un numero differente di immagini prive di target. Il confronto tra il Modello 1, addestrato senza immagini di sfondo, e il Modello 3, che include 29 immagini prive di target, evidenzia un peggioramento delle prestazioni di quest'ultimo. Analogamente il Modello 2, che non contiene frame di sfondo, risulta migliore sia del Modello 4, addestrato con 29 frame di sfondo, sia del Modello 5, addestrato con 11 frame di sfondo. Sebbene il Modello 5 presenti prestazioni superiori rispetto al Modello 4, entrambi risultano inferiori al Modello 2. Nel complesso, i risultati mostrano che l'introduzione di frame di sfondo non porta benefici in termini di prestazioni. Al contrario, all'aumentare del numero di immagini prive di target cresce la probabilità di mancate rilevazioni, riducendo la capacità della rete di identificare correttamente gli oggetti presenti nella scena.

Dai risultati ottenuti emerge che il Modello 2, addestrato sul dataset privo di frame di sfondo e utilizzando la Data Augmentation di tipo *2*, è quello che presenta le prestazioni migliori. In particolare, i valori lungo la diagonale principale della matrice risultano pari a 1 per quasi tutte le classi, con errori molto ridotti solamente

per le classi *orange_block* e *bottle*. Per la classe *orange_block* il valore sulla diagonale è pari a 0.99, indicando che il 99% dei campioni appartenenti a questa classe viene correttamente riconosciuto dal modello, mentre l'1% viene erroneamente classificato come *black-rectangle*. Per la classe *bottle*, invece, il valore sulla diagonale è pari a 0.91: questo significa che il 91% delle bottiglie viene identificato correttamente, mentre il 7% viene classificato come *black-rectangle* e il 2% non viene rilevato. In generale per tutti i modelli, gli errori evidenziati dalle matrici di confusione non sono riconducibili a una reale difficoltà del modello nel distinguere le diverse classi di oggetti. Infatti, quando il target viene rilevato, la classe assegnata dalla rete è corretta. Gli errori osservati sono invece riconducibili a mancate rilevazioni del target oppure a falsi positivi associati a oggetti presenti sullo sfondo, che vengono erroneamente identificati come una delle classi di interesse. Nella Figura 4.12 sono riportati due esempi di errori commessi dal Modello 2.

Figura 4.12: Esempi di errori commessi dal Modello 2 durante la valutazione sulle acquisizioni di Testing: mancato rilevamento di una bottle (in alto) e classificazione errata di un orange_block come black-rectangle (in basso).

4.6 Elaborazione delle Acquisizioni di Gaze

Una volta completata la selezione del modello migliore, si passa al secondo script Python dedicato all'elaborazione delle Acquisizioni di Gaze.

Come descritto in precedenza, per questa fase vengono utilizzati esclusivamente i video relativi alla palla blu e al blocco arancione, entrambi movimentati lungo una traiettoria ad arco. La scelta di considerare solamente questi due oggetti è legata ai risultati ottenuti durante la valutazione del Modello 2 sulle acquisizioni di Testing. In particolare, la palla blu ha ottenuto prestazioni ottime, con un valore pari a 1 lungo la diagonale principale della matrice di confusione, mentre il blocco arancione ha ottenuto prestazioni leggermente inferiori, con un valore pari a 0.99. Questa scelta consente di valutare l'intera procedura sia in una condizione ideale sia in una situazione leggermente più critica.

Lo script utilizza come dati di ingresso i pesi *best.pt* del Modello 2, i video delle Acquisizioni di Gaze esportati da Tobii Pro Glasses 3 senza la sovrapposizione del gaze, i corrispondenti file Excel esportati da Tobii Pro Lab contenenti i dati dello sguardo e i parametri delle rette di calibrazione ottenute nella fase precedente.

L'elaborazione si articola in quattro fasi principali. Nella prima fase viene effettuato il rilevamento e la localizzazione del target mediante YOLO. Successivamente vengono elaborati i dati di gaze esportati da Tobii Pro Lab. Nella terza fase i due flussi di dati vengono sincronizzati temporalmente, in modo da associare a ogni frame del video la corrispondente posizione dello sguardo. Infine, le coordinate ottenute vengono convertite in angoli visivi e utilizzate per determinare l'ampiezza del movimento oculare. Inoltre, ciascun frame è stato classificato come IN o OUT in base alla posizione dello sguardo rispetto al target: IN se il punto di sguardo ricade all'interno della bounding box del target, OUT in caso contrario.

4.6.1 Rilevamento del target mediante YOLO

Come prima operazione, la rete YOLO addestrata sul Modello 2 viene applicata ai video delle Acquisizioni di Gaze per localizzare il target all'interno di ciascun frame.

Come descritto in precedenza, l'output di questa elaborazione comprende:

- un file Excel contenente, per ogni frame, la classe dell'oggetto rilevato, il valore di confidenza, le coordinate della *bounding box* e la posizione del centroide;
- un video in cui vengono visualizzate le *bounding box* associate alle *detection* effettuate dalla rete.

Sebbene il Modello 2 abbia mostrato prestazioni elevate durante la fase di valutazione, la matrice di confusione evidenzia la presenza di un numero ridotto di errori di classificazione. Di conseguenza, la rete può comunque produrre occasionalmente rilevamenti errati. Per questo motivo sono state sviluppate due funzioni aggiuntive: la prima ha lo scopo di individuare ed eliminare le *detection* errate, mentre la seconda ricostruisce le informazioni mancanti mediante interpolazione lineare.

Rimozione dei falsi positivi

L'identificazione dei falsi positivi viene effettuata a partire dai file Excel generati dalla rete YOLO. Poiché in ogni acquisizione è presente un solo oggetto, nel file Excel dovrebbe comparire una sola classe.

La funzione analizza quindi tutte le righe del file e memorizza le classi rilevate. Se viene rilevata più di una classe, viene analizzata la durata della loro comparsa nel video. La classe presente soltanto per un numero limitato di frame viene considerata un falso positivo e le relative righe vengono eliminate dal file Excel. Nel video relativo alla palla blu non sono stati osservati falsi positivi.

Nel caso del blocco arancione, invece, sono state individuate due rilevazioni errate appartenenti a una classe diversa da quella dell'oggetto presente nella scena.

Le Figure 4.13 e 4.14 riportano le righe del file Excel relativo al video con blocco arancione prima e dopo la rimozione dei falsi positivi.

FRAME	CLASSE	CONFIDENZA	X1_MIN	Y1_MIN	X2_MAX	Y2_MAX	CENTROIDE_X	CENTROIDE_Y
96	orange block	0,7375	815	593	876	784	845	689
97	black-rectangle	0,7543	931	49	975	105	953	77
98	orange block	0,743	812	593	873	785	843	689

FRAME	CLASSE	CONFIDENZA	X1_MIN	Y1_MIN	X2_MAX	Y2_MAX	CENTROIDE_X	CENTROIDE_Y
897	orange block	0,7611	985	613	1051	801	1018	707
898	bottle	0,7608	862	49	903	137	883	93
899	orange block	0,7729	983	606	1049	803	1016	704

Figura 4.13: Rilevamenti errati presenti nel file Excel relativo al video con blocco arancione.

FRAME	CLASSE	CONFIDENZA	X1_MIN	Y1_MIN	X2_MAX	Y2_MAX	CENTROIDE_X	CENTROIDE_Y
96	orange block	0,7376	815	593	876	784	845	689
97								
98	orange block	0,7431	812	593	873	785	843	689

FRAME	CLASSE	CONFIDENZA	X1_MIN	Y1_MIN	X2_MAX	Y2_MAX	CENTROIDE_X	CENTROIDE_Y
897	orange block	0,7612	985	613	1051	801	1018	707
898								
899	orange block	0,7727	983	606	1049	803	1016	704

Figura 4.14: Risultato della procedura di pulizia del file Excel, con eliminazione dei falsi positivi individuati dalla rete YOLO.

Per verificare visivamente il corretto funzionamento della procedura, i falsi positivi identificati vengono evidenziati in rosso nel video di output, come mostrato in Figura 4.15.

Figura 4.15: Falsi positivi evidenziati in rosso nel video di output.

Interpolazione delle rilevazioni mancanti

Una volta rimossi i falsi positivi, alcuni frame risultano privi delle informazioni relative alla posizione dell'oggetto. Per ricostruire tali dati viene utilizzata una procedura di interpolazione lineare.

L'interpolazione lineare è una tecnica matematica che consente di stimare valori intermedi a partire da due campioni noti, assumendo una variazione lineare tra essi [52].Tale approccio risulta appropriato nel presente studio poiché il movimento del target avviene in modo continuo e regolare.

La funzione individua automaticamente i gruppi di frame mancanti presenti nella sequenza.

Per ciascun intervallo vengono considerati il frame valido precedente e quello successivo al gap; utilizzando questi due riferimenti vengono interpolati linearmente sia il centroide sia i vertici della *bounding box* mediante le seguenti relazioni:

$$x(t) = x_1 + \frac{t-t_1}{t_2-t_1}(x_2 - x_1) \qquad (3)$$

$$y(t) = y_1 + \frac{t-t_1}{t_2-t_1}(y_2 - y_1)$$

dove:

- (x_1, y_1) rappresentano le coordinate del frame valido precedente;
- (x_2, y_2) rappresentano le coordinate del frame valido successivo;
- t_1 e t_2 indicano gli istanti temporali dei due frame;
- t rappresenta l'istante temporale del frame da interpolare.

Al termine della procedura si ottengono:

- un file Excel aggiornato contenente la posizione reale dell'oggetto per ogni frame;
- un nuovo video di output nel quale sono riportate le *bounding box* corrette e le eventuali ricostruzioni ottenute tramite interpolazione.

Poiché la posizione dei frame interpolati non deriva direttamente da una rilevazione effettuata dalla rete YOLO, a tali frame non viene associato alcun valore di confidenza. Per facilitarne l'identificazione, essi vengono evidenziati in giallo sia nel file Excel sia nel video di output.

Le figure 4.16 e 4.17 mostrano i due frame ricostruiti sia nel file Excel sia nel video di output.

FRAME	CLASSE	CONFIDENZA	X1_MIN	Y1_MIN	X2_MAX	Y2_MAX	CENTROIDE_X	CENTROIDE_Y
96	orange block	0,7376	815	593	876	784	845	689
97	orange block		814	593	874	784	844	689
98	orange block	0,7431	812	593	873	785	843	689

FRAME	CLASSE	CONFIDENZA	X1_MIN	Y1_MIN	X2_MAX	Y2_MAX	CENTROIDE_X	CENTROIDE_Y
897	orange block	0,7612	985	613	1051	801	1018	707
898	orange block		984	610	1050	802	1017	706
899	orange block	0,7727	983	606	1049	803	1016	704

Figura 4.16: Frame ricostruiti tramite interpolazione lineare nel file Excel di output.

Figura 4.17: Visualizzazione delle bounding box associate ai frame ricostruiti mediante interpolazione lineare.

4.6.2 Elaborazione dei dati di gaze

La fase successiva riguarda l'elaborazione dei dati di gaze esportati da Tobii Pro Lab. Analizzando i file Excel si osserva la presenza di alcuni dati mancanti relativi alla posizione dello sguardo. Tali assenze possono essere dovute sia ai *blink* del soggetto sia a temporanee perdite di *tracking* da parte dell'Eye Tracker.

L'obiettivo era quindi distinguere queste due condizioni, così da poter eventualmente ricostruire tramite interpolazione soltanto i dati mancanti dovuti a errori di rilevamento. Per affrontare questo problema sono state valutate due differenti metodologie per l'identificazione automatica dei *blink.*

Identificazione dei blink mediante analisi dei gap temporali

La prima si basa sulla definizione classica di *blink*, inteso come perdita temporanea simultanea del tracciamento di entrambi gli occhi per una durata compresa tra 100 ms e 400 ms.

A tal fine, sono stati analizzati i dati *GazePointX e GazePointY* calcolando la durata temporale dei *gap* presenti nel segnale. Tuttavia, applicando questo criterio, il metodo tendeva a sottostimare il numero reale di *blink*.

In particolare, nel video relativo alla palla blu veniva identificato un solo *blink*, mentre nel video relativo al blocco arancione ne venivano rilevati soltanto due. Considerando che entrambe le acquisizioni hanno una durata di circa 40 secondi e che una persona sbatte mediamente le palpebre circa 17 volte al minuto, tali risultati non possono essere considerati realistici [53].

Identificazione dei blink mediante noise-based algorithm

È stata quindi valutata una seconda metodologia, proposta da Ronen Hershman, Avishai Henik e Noga Cohen nell'articolo *A novel blink detection method based on pupillometry noise* e denominata *noise-based algorithm*.

Questo approccio sfrutta il fatto che i dati registrati da un *Eye Tracker* presentano naturalmente piccole fluttuazioni ad alta frequenza dovute al rumore di misura dello strumento. Durante un *blink*, il movimento della palpebra altera temporaneamente il segnale, causando la scomparsa di tali fluttuazioni. L'algoritmo identifica quindi l'inizio e la fine del *blink* analizzando l'interruzione e la successiva ricomparsa del rumore di fondo.

Questo articolo propone un codice Python già implementato per l'identificazione automatica dei *blink* [54]. Tuttavia, il metodo originale è stato sviluppato considerando la presenza di un'unica variabile relativa alla dimensione della pupilla. Nei dati esportati da Tobii Pro Lab sono invece disponibili tre differenti segnali: *Pupil diameter left*, *Pupil diameter right* e *Pupil diameter filtered*. Per questo motivo il codice è stato adattato seguendo due approcci differenti. Nel primo caso l'algoritmo è stato applicato direttamente alla variabile *Pupil diameter filtered*, utilizzando un unico segnale pupillare. Nel secondo caso, invece, l'algoritmo è stato applicato separatamente ai dati dell'occhio destro e dell'occhio sinistro; successivamente i risultati sono stati combinati mediante un operatore logico AND, considerando valido un *blink* soltanto quando rilevato contemporaneamente in entrambi gli occhi. In entrambi i casi, tuttavia, l'algoritmo tende a classificare come *blink* quasi tutti i gap presenti nel segnale, inclusi intervalli di durata molto breve che risultano più plausibilmente associabili a temporanee perdite di tracking piuttosto che a reali ammiccamenti fisiologici.

Poiché nessuna delle metodologie analizzate ha consentito di distinguere in modo sufficientemente affidabile i *blink* dalle perdite temporanee di *tracking*, si è deciso

di non applicare alcuna procedura di interpolazione e di utilizzare direttamente i dati originali esportati da Tobii Pro Lab.

Di conseguenza, l'elaborazione del file Excel è stata limitata a una fase di selezione delle informazioni di interesse, analogamente a quanto effettuato per l'acquisizione di calibrazione, mantenendo esclusivamente le righe relative all'*Eye Tracker,* corrispondenti ai dati dello sguardo*,* e le colonne *Recording Timestamp*, *GazePointX* e *GazePointY*, che rappresentano rispettivamente il riferimento temporale di ciascun campione acquisito e le coordinate orizzontale e verticale del punto di sguardo.

4.6.3 Sincronizzazione temporale tra i video del target e dati di gaze

La fase successiva ha l'obiettivo di sincronizzare temporalmente i dati relativi alla posizione del target con quelli dello sguardo del soggetto. A tale scopo vengono utilizzati il file contenente la posizione del target ottenuta tramite YOLO e successivamente corretta mediante interpolazione e il file contenente i dati di gaze elaborati nella fase precedente.

L'obiettivo è associare a ogni frame del video di scena la corrispondente posizione dello sguardo. Tuttavia, i due segnali vengono acquisiti con frequenze differenti: il video della scena osservata è registrato a 25 frame al secondo, mentre il segnale di *gaze* viene campionato a circa 100 Hz, di conseguenza, è necessario sincronizzare temporalmente i due segnali.

Come prima operazione, a ciascun frame del video viene associato il relativo *timestamp* espresso in millisecondi, utilizzando la stessa unità temporale dei dati di *gaze*. Il *timestamp* viene calcolato mediante la seguente relazione:

$$Timestamp_{frame} = \frac{(frame - 1) \times 1000}{FPS}$$

dove $frame$ rappresenta il numero progressivo del frame, FPS rappresenta il frame rate di 25 frame/s. Il termine $(frame - 1)$ viene utilizzato affinché il primo

frame corrisponda al *timestamp* iniziale pari a 0 ms, mentre il fattore $\times 1000$ consente di convertire il tempo da secondi a millisecondi.

Successivamente, per ogni frame del video viene considerato il relativo *timestamp* e vengono ricercati nel file Excel contenente i dati di gaze i due campioni temporalmente più vicini ossia il campione con *timestamp* immediatamente precedente e il campione con *timestamp* immediatamente successivo. Utilizzando questi due campioni viene applicata la procedura di interpolazione lineare descritta precedentemente (3), così da stimare i valori *GazePointX e GazePointY* corrispondenti all'istante temporale del *frame*.

Nel codice sono stati inoltre gestiti alcuni casi limite:

- se il *timestamp* del frame precede il primo campione disponibile, viene utilizzato direttamente il primo valore del gaze;
- se il *timestamp* è successivo all'ultimo campione disponibile, viene utilizzato l'ultimo valore presente nel segnale.

È stata inoltre implementata una gestione dei dati mancanti: se uno dei due campioni di gaze utilizzati per l'interpolazione presenta un valore mancante, l'interpolazione non viene eseguita e il corrispondente valore di *gaze* viene mantenuto vuoto.

Al termine della procedura si ottiene un unico file Excel sincronizzato contenente, per ogni frame del video, sia la posizione del target sia la corrispondente posizione dello sguardo del soggetto (Figura 4.18).

		Posizione Occhio		Posizione Target					
FRAME	TIMESTAMP_MS	GAZE_X_INTERPOLATO	GAZE_Y_INTERPOLATO	X1_MIN	Y1_MIN	X2_MAX	Y2_MAX	CENTROIDE_X	CENTROIDE_Y
1	0	1038	648	999	574	1058	765	1028	670
2	40	1028	636	999	574	1058	765	1028	670
3	80	1035	639	999	573	1058	765	1028	669
4	120	1033	638	1001	574	1059	765	1030	670
5	160	1032	638	1002	575	1061	766	1032	670
6	200	1031	638	1004	575	1063	767	1034	671
7	240	1030	639	1004	575	1063	767	1034	671

Figura 4.18: Prime righe dell'Excel sincronizzato.

4.6.4 Calcolo degli angoli oculari e classificazione IN/OUT

L'ultima fase dell'elaborazione prevede due elaborazioni distinte. La prima consiste nella conversione delle coordinate del target e dello sguardo da pixel a gradi visivi e nel calcolo della distanza angolare tra i due punti. La seconda consiste nella verifica, per ciascun frame, che il punto di sguardo ricada all'interno della *bounding box* del target, così da identificare i momenti in cui il soggetto sta osservando l'oggetto di interesse classificandoli come IN.

L'analisi viene eseguita a partire dal file Excel sincronizzato ottenuto nella fase precedente.

Calcolo angoli oculari

Per il calcolo degli angoli oculari vengono applicate le rette di calibrazione determinate durante l'elaborazione dell'acquisizione di calibrazione.

Le rette di calibrazione vengono applicate sia alle coordinate del punto di sguardo (*gaze_x*, *gaze_y*) sia alle coordinate del centroide del target (*centroide_x*, *centroide_y*) presenti nel file Excel sincronizzato. Per tale conversione vengono utilizzate le rette calcolate nella fase di calibrazione descritta in precedenza. Le due acquisizioni di gaze considerate utilizzano tali rette, ma per eventuali nuove acquisizioni, esse dovranno essere ricalcolate mediante una nuova procedura di calibrazione. Nel caso in esame, le equazioni sono le seguenti:

$$\vartheta_{X\ stimato} = 0.054x - 53.439$$

$$\vartheta_{Y\ stimato} = -0.058y + 21.136$$

L'applicazione di tali equazioni permette di ottenere, per ciascun frame, gli angoli associati alla direzione dello sguardo e alla posizione del target.

Una volta determinati gli angoli, viene calcolata la distanza angolare tra il punto di sguardo e il target, al fine di quantificare quanto lo sguardo del soggetto sia distante dall'oggetto di interesse.

La distanza viene calcolata tramite la seguente formula:

$$d = \sqrt{(x_{gaze} - x_{target})^2 + (y_{gaze} - y_{target})^2}$$

dove $x_{gaze}\, e\, y_{gaze}$ rappresentano gli angoli oculari dello sguardo, mentre $x_{target}\, e\, y_{target}$ indicano gli angoli relativi alla posizione del target.

È importante sottolineare che gli angoli ottenuti rappresentano angoli relativi rispetto alla posizione della testa e non angoli assoluti nello spazio. Per ottenere una misura assoluta dello sguardo sarebbe infatti necessario considerare anche i movimenti della testa durante l'acquisizione.

Classificazione IN/OUT

L'analisi successiva è finalizzata a determinare se il soggetto stia osservando il target rilevato da YOLO. A tal fine, per ogni frame viene verificato se il punto di gaze ricade all'interno della *bounding box* associata al target. In base a questa verifica, ciascun frame viene classificato come:

- IN, se il punto di gaze si trova all'interno della *bounding box*;
- OUT, se il punto di gaze ricade all'esterno della *bounding box*.

Questa classificazione, tuttavia, risulta particolarmente sensibile a piccoli errori di localizzazione. Infatti, anche un lieve scostamento del punto di gaze può determinare la classificazione di un frame come OUT, pur essendo lo sguardo molto vicino al target. Per questo motivo è stata eseguita una seconda analisi introducendo un margine di tolleranza (*padding*) attorno alla *bounding box*. In particolare, la *bounding box* originale è stata ampliata di 11 pixel per lato, valore ottenuto convertendo in pixel l'accuratezza dichiarata dell'*Eye Tracker* (0.6°), come descritto in precedenza (2). In questo modo è stato possibile confrontare i risultati della classificazione con e senza l'applicazione del margine di tolleranza.

Capitolo 5: Risultati finali

5.1 Analisi della classificazione IN/OUT

L'algoritmo di classificazione IN/OUT produce come output un file Excel avente la stessa struttura del file sincronizzato utilizzato in ingresso, al quale viene aggiunta una colonna contenente la classificazione associata a ciascun frame. Inoltre, viene generato un video di output in cui sono visualizzati il punto di gaze, la *bounding box* del target e la relativa classificazione IN/OUT.

L'analisi è stata eseguita per entrambe le acquisizioni considerate, sia utilizzando la *bounding box* originale sia introducendo il *padding* descritto nel capitolo precedente. Nei video di output la *bounding box* originale è rappresentata in giallo, mentre la *bounding box* estesa mediante *padding* è evidenziata in verde.

Come descritto in precedenza, durante le acquisizioni il soggetto era istruito a osservare il target solamente quando questo transitava nella zona centrale della scena, evitando invece di seguirlo durante gli spostamenti verso le regioni laterali. Di conseguenza, l'andamento atteso della classificazione può essere suddiviso in cinque fasi principali:

- una prima fase IN, corrispondente all'osservazione del target nella zona centrale;
- una fase OUT durante lo spostamento verso sinistra;
- una seconda fase IN durante il ritorno del target verso il centro;
- una nuova fase OUT durante lo spostamento verso destra;
- una fase finale IN durante il successivo ritorno nella zona centrale.

L'analisi delle sequenze temporali ottenute utilizzando la *bounding box* senza *padding* conferma la presenza delle cinque fasi attese. Tuttavia, i periodi classificati come IN non risultano completamente continui, ma presentano

occasionali frame classificati come OUT, mentre le fasi OUT risultano generalmente continue e prive di interruzioni.

Nelle Figure 5.1 e 5.2 sono mostrati alcuni esempi rappresentativi di sequenze classificate come IN e come OUT.

		Posizione Occhio		Posizione Target						
FRAME	TIMESTAMP_MS	GAZE_X_INTERPOLATO	GAZE_Y_INTERPOLATO	X1_MIN	Y1_MIN	X2_MAX	Y2_MAX	CENTROIDE_X	CENTROIDE_Y	IN/OUT
910	36425	972	710	937	649	1023	745	980	697	IN
911	36465			938	648	1023	745	980	696	
912	36505			936	648	1022	744	979	696	
913	36545			935	647	1021	743	978	695	
914	36585			935	647	1021	743	978	695	
915	36625			936	648	1021	743	978	696	
916	36665			937	648	1022	744	980	696	
917	36706	984	836	938	649	1022	744	980	696	OUT
918	36746	986	711	937	649	1022	744	980	696	IN
919	36786	986	684	938	649	1022	744	980	696	IN
920	36826	981	683	937	649	1023	744	980	696	IN
921	36866	981	684	937	649	1023	744	980	696	IN
922	36906	978	687	938	650	1024	745	981	698	IN
923	36946	979	684	937	649	1023	744	980	696	IN
924	36986	978	684	937	649	1023	744	980	696	IN
925	37026	979	684	938	649	1025	744	982	696	IN
926	37066	980	685	938	647	1025	744	982	696	IN
927	37106	980	686	939	646	1025	744	982	695	IN
928	37146	982	685	938	646	1025	744	982	695	IN
929	37186	981	686	938	645	1024	744	981	694	IN
930	37227	982	684	937	645	1024	742	980	694	IN
931	37267	980	684	937	645	1024	742	980	694	IN
932	37307	980	684	938	644	1024	742	981	693	IN
933	37347	980	684	939	643	1025	742	982	692	IN
934	37387	980	682	938	643	1025	741	982	692	IN
935	37427	980	682	938	643	1024	742	981	692	IN
936	37467	978	681	937	644	1023	743	980	694	IN
937	37507	977	681	938	644	1023	743	980	694	IN
938	37547	956	687	938	644	1024	743	981	694	IN
939	37587	937	692	938	644	1025	742	982	693	OUT
940	37627	936	690	938	644	1024	742	981	693	OUT
941	37667	935	691	937	644	1024	742	980	693	OUT
942	37707	935	690	936	643	1023	741	980	692	OUT
943	37747	934	690	936	642	1022	740	979	691	OUT
944	37788	933	690	935	643	1022	740	978	692	OUT
945	37828	934	690	936	643	1022	740	979	692	OUT
946	37868	974	678	938	643	1023	740	980	692	IN

Figura 5.1: Esempio di sequenza classificata prevalentemente come IN, con presenza di frame OUT isolati, analizzata senza padding.

		Posizione Occhio		Posizione Target						
FRAME	TIMESTAMP_MS	GAZE_X_INTERPOLATO	GAZE_Y_INTERPOLATO	X1_MIN	Y1_MIN	X2_MAX	Y2_MAX	CENTROIDE_X	CENTROIDE_Y	IN/OUT
320	12783	815	448	346	835	463	958	404	896	OUT
321	12823	813	436	350	837	463	957	406	897	OUT
322	12863	814	437	352	836	464	957	408	896	OUT
323	12903	816	440	351	835	464	957	408	896	OUT
324	12943	817	444	349	835	463	956	406	896	OUT
325	12983	831	444	350	835	463	957	406	896	OUT
326	13023	834	442	352	835	464	956	408	896	OUT
327	13063	835	443	353	834	466	953	410	894	OUT
328	13103	837	444	353	833	466	952	410	892	OUT
329	13143	840	444	354	832	467	952	410	892	OUT
330	13183	842	444	353	831	467	953	410	892	OUT
331	13223	808	442	353	831	467	953	410	892	OUT
332	13263	794	417	352	831	466	952	409	892	OUT

Figura 5.2: Esempio di sequenza continua classificata come OUT, analizzata senza padding.

Analizzando i periodi classificati come IN, è possibile distinguere due diverse tipologie di frame OUT. Alcuni sono localizzati in prossimità di gap nei dati dell'*Eye Tracker*, mentre altri compaiono come eventi all'interno di sequenze continue di frame IN.

Come discusso in precedenza, i gap possono essere dovuti ai blink del soggetto oppure a temporanee perdite di segnale dell'*Eye Tracker*. In entrambe le situazioni la qualità del tracciamento può risultare compromessa anche nei frame immediatamente precedenti e successivi al gap, generando stime meno accurate della posizione del gaze [53]. Gli OUT osservati in prossimità dei gap potrebbero quindi essere riconducibili a tali fenomeni.

Gli OUT presenti all'interno di sequenze continue di frame classificati come IN sono invece probabilmente dovuti a piccoli errori di localizzazione del punto di gaze oppure al fatto che il soggetto stesse osservando una regione molto vicina al bordo della *bounding box* del target.

Per valutare quantitativamente l'effetto dell'introduzione del *padding*, è stata innanzitutto calcolata la percentuale di frame classificati come IN nelle diverse condizioni analizzate. I risultati mostrano che l'applicazione del *padding* aumenta il numero di frame classificati correttamente come IN, correggendo le classificazioni OUT errate presenti all'interno di sequenze di frame IN. Nel dettaglio, per l'acquisizione con la palla blu tale percentuale passa dal 38.27% al 40.35%, mentre per l'acquisizione con il blocco arancione aumenta dal 31.33% al 35.55%.

Per comprendere quali frame fossero stati riclassificati, sono stati successivamente confrontati i file Excel ottenuti con e senza l'applicazione del margine di tolleranza. Dal confronto emerge che il *padding* riclassifica come IN alcuni frame precedentemente classificati come OUT all'interno delle sequenze di frame IN, mentre gli OUT localizzati in corrispondenza dei gap mantengono generalmente la classificazione originaria. La Figura 5.3 illustra questa differenza, riportando gli stessi frame mostrati in Figura 5.1 dopo l'applicazione del *padding.*

		Posizione Occhio		Posizione Target						
FRAME	TIMESTAMP_MS	GAZE_X_INTERPOLATO	GAZE_Y_INTERPOLATO	X1_MIN	Y1_MIN	X2_MAX	Y2_MAX	CENTROIDE_X	CENTROIDE_Y	IN/OUT
910	36425	972	710	937	649	1023	745	980	697	IN
911	36465			938	648	1023	745	980	696	
912	36505			936	648	1022	744	979	696	
913	36545			935	647	1021	743	978	695	
914	36585			935	647	1021	743	978	695	
915	36625			936	648	1021	743	978	696	
916	36665			937	648	1022	744	980	696	
917	36706	984	836	938	649	1022	744	980	696	OUT
918	36746	986	711	937	649	1022	744	980	696	IN
919	36786	986	684	938	649	1022	744	980	696	IN
920	36826	981	683	937	649	1023	744	980	696	IN
921	36866	981	684	937	649	1023	744	980	696	IN
922	36906	978	687	938	650	1024	745	981	698	IN
923	36946	979	684	937	649	1023	744	980	696	IN
924	36986	978	684	937	649	1023	744	980	696	IN
925	37026	979	684	938	649	1025	744	982	696	IN
926	37066	980	685	938	647	1025	744	982	696	IN
927	37106	980	686	939	646	1025	744	982	695	IN
928	37146	982	685	938	646	1025	744	982	695	IN
929	37186	981	686	938	645	1024	744	981	694	IN
930	37227	982	684	937	645	1024	742	980	694	IN
931	37267	980	684	937	645	1024	742	980	694	IN
932	37307	980	684	938	644	1024	742	981	693	IN
933	37347	980	684	939	643	1025	742	982	692	IN
934	37387	980	682	938	643	1025	741	982	692	IN
935	37427	980	682	938	643	1024	742	981	692	IN
936	37467	978	681	937	644	1023	743	980	694	IN
937	37507	977	681	938	644	1023	743	980	694	IN
938	37547	956	687	938	644	1024	743	981	694	IN
939	37587	937	692	938	644	1025	742	982	693	IN
940	37627	936	690	938	644	1024	742	981	693	IN
941	37667	935	691	937	644	1024	742	980	693	IN
942	37707	935	690	936	643	1023	741	980	692	IN
943	37747	934	690	936	642	1022	740	979	691	IN
944	37788	933	690	935	643	1022	740	978	692	IN
945	37828	934	690	936	643	1022	740	979	692	IN
946	37868	974	678	938	643	1023	740	980	692	IN

Figura 5.3: Stessi frame riportati in Figura 5.1, ma analizzati utilizzando la bounding box con padding

Per comprendere il motivo di questa diversa riclassificazione, sono stati analizzati i video di output relativi ai due casi. Le Figure 5.4 e 5.5 riportano due esempi rappresentativi.

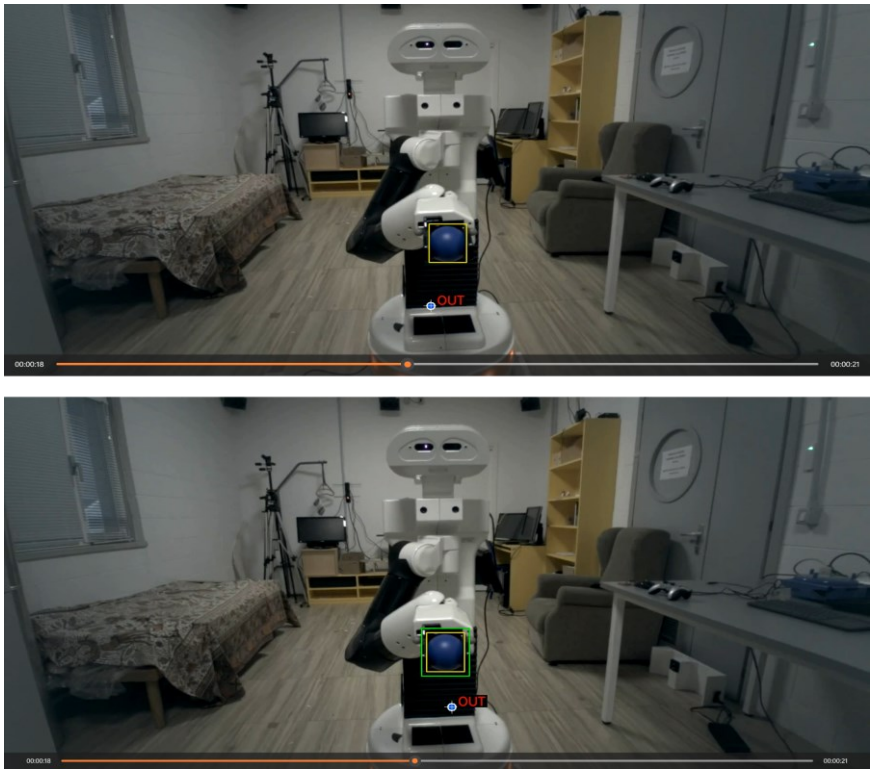

Figura 5.4: Confronto dello stesso frame senza padding (in alto) e con padding (in basso). In questo caso la classificazione rimane OUT anche dopo l'applicazione del padding.

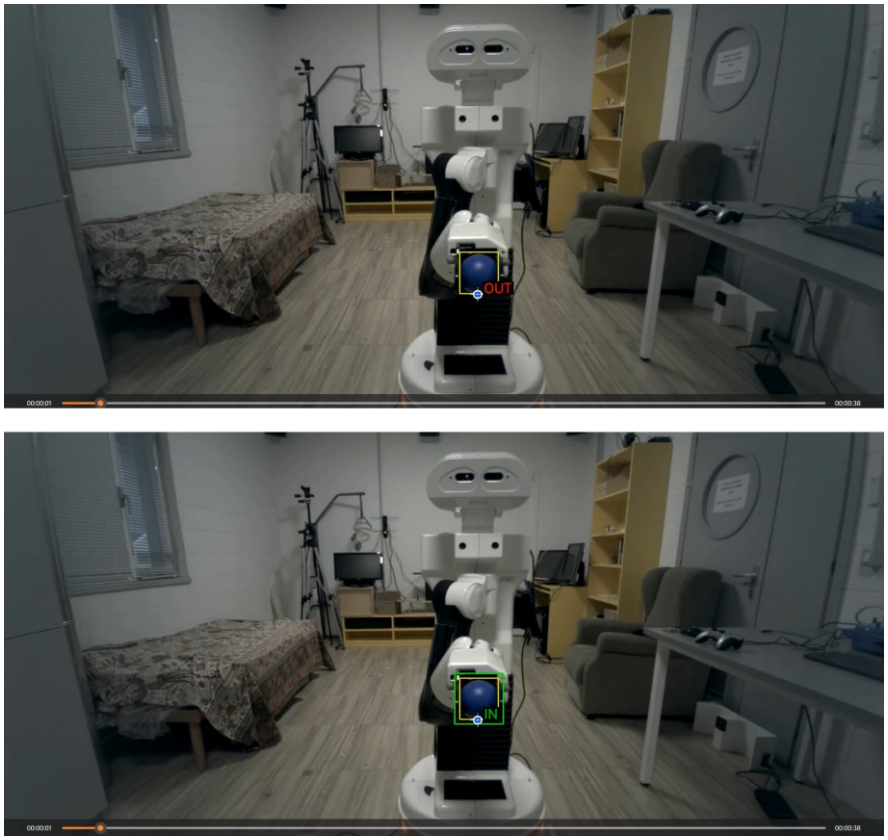

Figura 5.5: Confronto dello stesso frame senza padding (in alto) e con padding (in basso). In questo caso, l'applicazione del padding modifica la classificazione del frame da OUT a IN.

Nel primo caso è mostrato un frame classificato come OUT in prossimità di un gap nei dati. In questa situazione il punto di gaze risulta molto distante dal target e, di conseguenza, l'ampliamento della *bounding box* mediante *padding* non è sufficiente a modificarne la classificazione. Nel secondo caso, invece, è riportato un frame OUT appartenente a una sequenza continua di frame IN. Qui il punto di gaze si trova in prossimità del bordo della *bounding box* del target e l'introduzione del *padding* risulta sufficiente a riclassificare il frame come IN.

Questi esempi mostrano come il *padding* sia efficace nel riclassificare gli OUT dovuti a piccoli errori di localizzazione del gaze, mentre non sia sufficiente a modificare la classificazione degli OUT associati ai gap, nei quali il punto di gaze risulta generalmente più distante dal target.

5.2 Conversione dei valori da pixel ad angoli

L'algoritmo di conversione genera come output un file Excel contenente, per ciascun frame, le coordinate del target e del punto di gaze sia in pixel sia in gradi, oltre alla distanza angolare tra i due punti. Le prime righe dei file Excel ottenuti per le due acquisizioni analizzate sono riportate nella Figura 5.6.

		GAZE ORIGINALI		ANGOLI GAZE		CENTROIDI ORIGINALI		ANGOLI CENTROIDI		DISTANZA ANGOLARE	STATISTICHE	
FRAME	TIMESTAMP	GAZE_X_INTERPOLATO	GAZE_Y_INTERPOLATO	GAZE_X_ANGOLO	GAZE_Y_ANGOLO	CENTROIDE_X	CENTROIDE_Y	CENTROIDE_X_ANGOLO	CENTROIDE_Y_ANGOLO	GRADI	TIPO	VALORE
1	0	1020	615	1,641	-14,534	1025	618	1,911	-14,708	0,32121	ANGOLO MASSIMO	66,6259
2	40	1020	614	1,641	-14,476	1025	620	1,911	-14,824	0,44046	ANGOLO MINIMO	0,07925
3	80	1020	615	1,641	-14,534	1026	620	1,965	-14,824	0,43483		
4	120	1020	617	1,641	-14,65	1026	620	1,965	-14,824	0,36777		
5	160	1020	618	1,641	-14,708	1026	620	1,965	-14,824	0,34414		
6	200	1022	618	1,749	-14,708	1027	620	2,019	-14,824	0,29386		
7	240	1023	617	1,803	-14,65	1027	620	2,019	-14,824	0,27737		
8	280	1024	618	1,857	-14,708	1028	618	2,073	-14,708	0,216		
9	320	1022	618	1,749	-14,708	1028	617	2,073	-14,65	0,32915		
10	360	1021	616	1,695	-14,592	1026	618	1,965	-14,708	0,29386		
11	400	1022	614	1,749	-14,476	1026	618	1,965	-14,708	0,31699		
12	440					1028	617	2,073	-14,65			
13	480					1028	618	2,073	-14,708			
14	520	1032	736	2,289	-21,552	1028	618	2,073	-14,708	6,84741		

		GAZE ORIGINALI		ANGOLI GAZE		CENTROIDI ORIGINALI		ANGOLI CENTROIDI		DISTANZA ANGOLARE	STATISTICHE	
FRAME	TIMESTAMP	GAZE_X_INTERPOLATO	GAZE_Y_INTERPOLATO	GAZE_X_ANGOLO	GAZE_Y_ANGOLO	CENTROIDE_X	CENTROIDE_Y	CENTROIDE_X_ANGOLO	CENTROIDE_Y_ANGOLO	GRADI	TIPO	VALORE
1	0	1038	648	2,613	-16,448	1028	670	2,073	-17,724	1,38556	ANGOLO MASSIMO	55,80077
2	40	1028	636	2,073	-15,752	1028	670	2,073	-17,724	1,972	ANGOLO MINIMO	0,309458
3	80	1035	639	2,451	-15,926	1028	669	2,073	-17,666	1,780585		
4	120	1033	638	2,343	-15,868	1030	670	2,181	-17,724	1,863057		
5	160	1032	638	2,289	-15,868	1032	670	2,289	-17,724	1,856		
6	200	1031	638	2,235	-15,868	1034	671	2,397	-17,782	1,920844		
7	240	1030	639	2,181	-15,926	1034	671	2,397	-17,782	1,868527		
8	280	1030	639	2,181	-15,926	1034	671	2,397	-17,782	1,868527		
9	320	1030	640	2,181	-15,984	1034	670	2,397	-17,724	1,753356		
10	360	1028	642	2,073	-16,1	1034	670	2,397	-17,724	1,656005		
11	400	1025	646	1,911	-16,332	1036	670	2,505	-17,724	1,51344		
12	440	1023	650	1,803	-16,564	1036	671	2,505	-17,782	1,405819		
13	480	1024	649	1,857	-16,506	1035	670	2,451	-17,724	1,355124		
14	520	1024	649	1,857	-16,506	1034	670	2,397	-17,724	1,332338		

Figura 5.6: Prime righe dei file Excel ottenuti dall'analisi. In alto è riportato il file relativo all'acquisizione con la palla blu, mentre in basso quello relativo all'acquisizione con il blocco arancione.

A partire dalla distanza angolare sono stati inoltre determinati i valori minimo e massimo registrati durante ciascuna acquisizione. Per l'acquisizione con la palla blu si osserva una distanza angolare minima di circa 0.08° e una massima di circa 66.62°, mentre per l'acquisizione con il blocco arancione i valori risultano pari rispettivamente a 0.31° e 55.80°. Questi risultati forniscono un'indicazione del range di distanze angolari osservato durante le acquisizioni e costituiscono la base per l'analisi successiva, nella quale la distanza angolare viene confrontata con la classificazione IN/OUT.

5.3 Analisi congiunta

Per approfondire l'effetto dell'introduzione del *padding*, sono stati calcolati i valori minimo, massimo e medio della distanza angolare separatamente per i frame classificati come IN e come OUT. I risultati ottenuti per le due acquisizioni sono riportati nelle Tabelle 5.1 e 5.2.

Tabella 5.1: Statistiche della distanza angolare per l'acquisizione della palla blu, con e senza padding.

	Classificazione	**Angolo Min (°)**	**Angolo Max (°)**	**Angolo Medio (°)**
Acquisizione Palla Blu- senza padding	IN	0.079	3.524	0.99
	OUT	2.431	66.626	39.384
Acquisizione Palla Blu- con padding	IN	0.079	4.511	1.0975
	OUT	3.422	66.626	40,655

Tabella 5.1: Statistiche della distanza angolare per l'acquisizione del blocco arancione, con e senza padding.

	Classificazione	**Angolo Min (°)**	**Angolo Max (°)**	**Angolo Medio (°)**
Acquisizione Blocco Arancione- senza padding	IN	0.309	5.387	2.121
	OUT	1.945	55.801	34.04
Acquisizione Blocco Arancione- con padding	IN	0.309	5.387	2.174
	OUT	2.485	55.801	36.104

Per l'acquisizione con la palla blu, riportata nella Tabella 5.1, l'introduzione del *padding* determina un aumento sia dell'angolo massimo classificato come IN, che passa da 3.52° a 4.51°, sia dell'angolo minimo classificato come OUT, che passa da 2.43° a 3.42°. Ciò significa che, dopo l'applicazione del *padding*, alcuni frame che senza *padding* sarebbero stati classificati come OUT vengono invece classificati come IN. Questo comportamento è illustrato nelle Figure 5.7 e 5.8.

La Figura 5.7 confronta i frame corrispondenti all'angolo massimo classificato come IN nelle configurazioni senza e con *padding*, mostrando come quest'ultimo consenta di mantenere la classificazione IN anche per una maggiore distanza angolare tra il punto di gaze e il target.

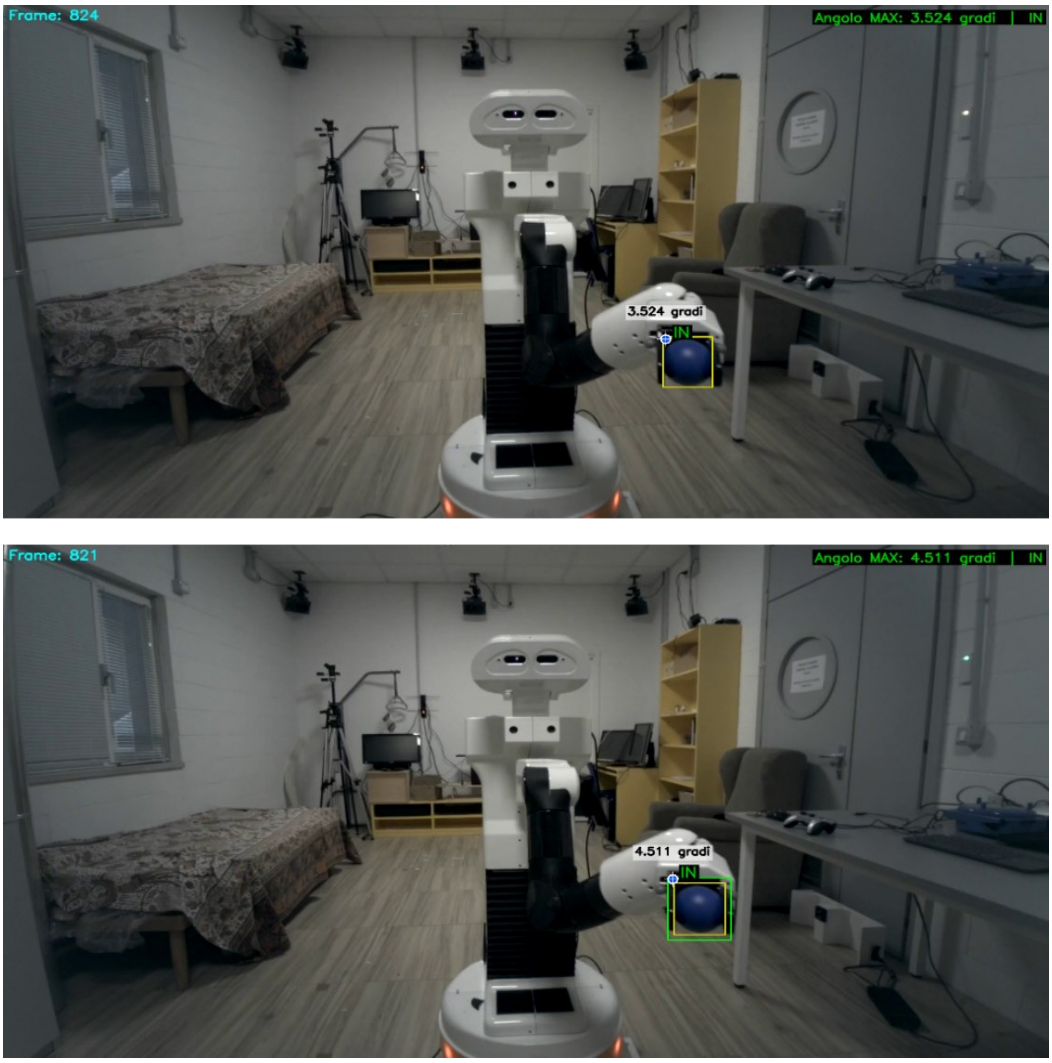

Figura 5.7: Confronto tra i frame corrispondenti all'angolo massimo classificato come IN nelle configurazioni senza padding (in alto) e con padding (in basso).

La Figura 5.8, invece, riporta i frame corrispondenti all'angolo minimo classificato come OUT nelle due configurazioni, evidenziando come, con il *padding*, sia necessario un angolo maggiore affinché un frame venga classificato come OUT.

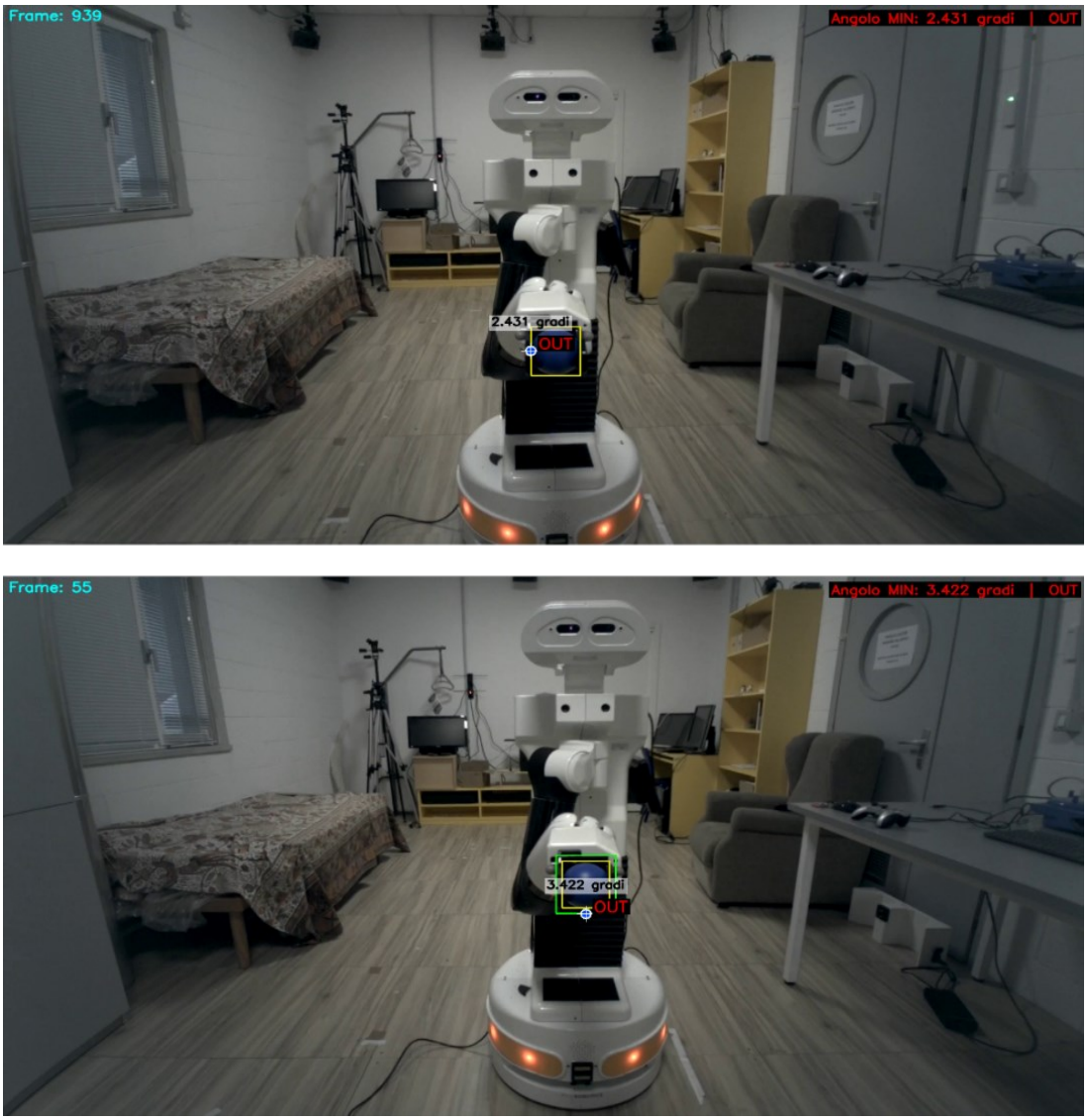

Figura 5.8: Confronto tra i frame corrispondenti all'angolo minimo classificato come OUT nelle configurazioni senza padding (in alto) e con padding (in basso).

Per quanto riguarda l'acquisizione con il blocco arancione, come riportato nella Tabella 5.2, l'introduzione del *padding* non modifica l'angolo massimo classificato come IN, che rimane pari a 5.38°. Si osserva invece un incremento dell'angolo minimo classificato come OUT, che passa da 1.94° a 2.48°. Questo risultato indica che, in questa acquisizione, il *padding* non riclassifica come IN frame caratterizzati da distanze angolari maggiori, ma aumenta la distanza angolare minima a partire dalla quale un frame viene classificato come OUT.

La Figura 5.9 riporta i frame corrispondenti all'angolo minimo classificato come OUT nelle configurazioni senza e con *padding*.

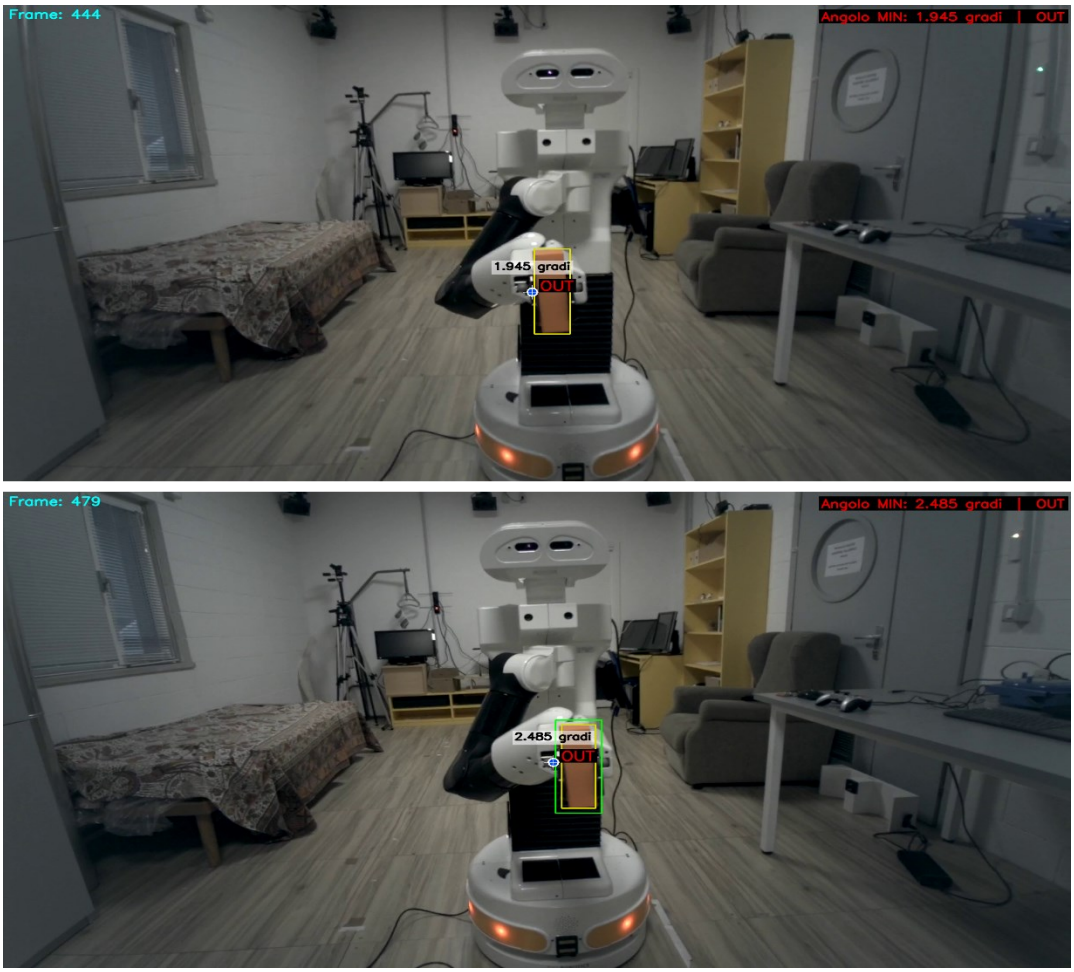

Figura 5.9: Confronto tra i frame corrispondenti all'angolo minimo classificato come OUT nelle configurazioni senza padding (in alto) e con padding (in basso).

5.3 Tempi di esecuzione

Come ultima analisi della tesi, sono stati valutati i tempi di elaborazione della pipeline Python sviluppata per l'analisi delle acquisizioni di gaze, considerando ciascuno degli step che la compongono. Il calcolo è stato effettuato separatamente per le due acquisizioni.

Come mostrato nella Tabella 5.3, i primi cinque step della pipeline richiedono gli stessi tempi di esecuzione per entrambe le acquisizioni, per un tempo complessivo di circa 72 s. In questa fase vengono eseguiti il rilevamento del target mediante YOLO, la rimozione dei falsi positivi, l'interpolazione lineare, la sincronizzazione tra i dati di gaze e il video della scena e il calcolo degli angoli del punto di gaze e del target.

Gli step successivi sono stati invece valutati sia senza *padding* sia con *padding*. Come si può osservare, l'aggiunta del *padding* comporta solo un leggero aumento dei tempi di elaborazione. Nel complesso, il tempo totale della pipeline rimane inferiore a 2 minuti per l'acquisizione con la palla blu e a circa 2 minuti e 30 secondi per quella con il blocco arancione.

Per quanto riguarda le risorse hardware utilizzate, la GPU è stata impiegata esclusivamente durante la fase di inferenza del modello YOLO, mentre tutte le elaborazioni successive sono state eseguite sulla CPU.

Tabella 5.2: Tempi di elaborazione dei vari step della pipeline Python per le acquisizioni di Gaze.

	Acquisizione con Palla Blu		**Acquisizione con Blocco Arancione**	
Step1: Rilevamento target (YOLO)	45 s		45 s	
Step 2: Rimozione falsi positivi	11s		11s	
Step 3: Interpolazione lineare	11s		11s	
Step 4: Sincronizzazione gaze-video	5 s		5 s	
Step 5: Calcolo degli angoli di gaze e target	0.20 s		0.20 s	
Step 6: Classificazione IN/OUT	**No Padding**	**Si padding**	**No Padding**	**Si padding**
	0.38 s	0.47 s	0.49 s	0.53 s
Step 7: Statistiche angolari IN/OUT	0.03 s	0.04s	0.06 s	0.05 s
Step 8: Creazione Video IN/OUT	17 s	19s	29 s	30 s
Step 9: Creazione Video IN/OUT + Angoli	22 s	24 s	32 s	40 s
Step 10: Salvataggio frame significativi	1 s	1s	1 s	1 s
Tempi Totali	**1 min 53 s**	**1 min 57 s**	**2 min 15 s**	**2 min 24 s**

Conclusioni

L'obiettivo di questo lavoro di tesi è stato lo sviluppo di una pipeline per la valutazione quantitativa di esercizi di *Smooth Pursuit Training* mediante l'utilizzo di un sistema di *Eye Tracking* Tobii Pro Glasses 3 e di un sistema robotico TIAGo. In particolare, il robot è stato utilizzato per movimentare il bersaglio lungo traiettorie prestabilite, mentre l'*Eye Tracker* ha consentito di registrare i movimenti oculari del soggetto durante l'esecuzione dell'esercizio.

La pipeline sviluppata, basata su un modello YOLO appositamente addestrato e su una serie di elaborazioni implementate in Python, consente di ottenere per ogni frame le coordinate angolari del punto di gaze e del target, la distanza angolare tra le due posizioni e la relativa classificazione IN/OUT. Le informazioni ottenute rappresentano parametri quantitativi utili per la valutazione delle prestazioni del soggetto, permettendo di analizzare la capacità di mantenere l'inseguimento del target nel tempo e di individuare eventuali difficoltà o limitazioni nell'esplorazione visiva.

Nonostante i risultati ottenuti, il sistema presenta ancora alcune limitazioni. La prima riguarda il livello di automazione complessivo del processo: sebbene gran parte delle elaborazioni sia eseguita automaticamente, la fase di acquisizione richiede ancora l'intervento dell'operatore per l'avvio e l'arresto delle registrazioni e per l'esportazione dei dati necessari all'analisi. Inoltre, la procedura di calibrazione sviluppata in MATLAB richiede la selezione manuale degli intervalli corrispondenti ai diversi gradini di calibrazione per il calcolo delle mediane utilizzate nella stima delle rette di conversione.

Un secondo aspetto riguarda la modalità di esecuzione della calibrazione: attualmente la griglia viene mantenuta manualmente dall'operatore e potrebbe quindi non risultare perfettamente perpendicolare all'asse visivo del soggetto, introducendo una potenziale fonte di errore nella stima delle rette di calibrazione.

Un’ulteriore limitazione è rappresentata dalla presenza di dati mancanti nei segnali di gaze esportati da Tobii Pro Lab. Durante il lavoro sono stati valutati diversi approcci per distinguere i dati mancanti dovuti ai blink fisiologici da quelli causati da temporanee perdite di tracking dell’*Eye Tracker*. Tuttavia, non è stato possibile individuare una metodologia sufficientemente affidabile per discriminare automaticamente le due condizioni e ricostruire i campioni mancanti.

I principali sviluppi futuri riguardano quindi il superamento delle limitazioni attualmente presenti. In primo luogo, le procedure attualmente implementate in MATLAB e Python potrebbero essere integrate in un'unica applicazione software, al fine di semplificare il flusso di lavoro e ridurre ulteriormente l'intervento richiesto all'operatore. Parallelamente, la procedura di calibrazione potrebbe essere migliorata mediante l'impiego di un supporto dedicato, come un cavalletto, in grado di mantenere la griglia in una posizione stabile e perpendicolare all'asse visivo del soggetto, riducendo una potenziale fonte di errore nella stima delle rette di conversione. Un ulteriore sviluppo riguarda l'esecuzione dell'intera pipeline in tempo reale, consentendo il calcolo online delle metriche durante lo svolgimento dell'esercizio. Inoltre, l'introduzione di nuove metriche potrebbe fornire una caratterizzazione più completa del comportamento oculomotorio del soggetto. Infine, sarà fondamentale validare il sistema su pazienti affetti da eminattenzione spaziale, al fine di valutarne l'efficacia e il potenziale impiego come strumento riabilitativo.

Nel complesso, il lavoro svolto ha permesso di verificare la validità dell'approccio proposto e di realizzare una pipeline in grado di fornire misure quantitative utili per l'analisi degli esercizi di *Smooth Pursuit Training.* I risultati ottenuti rappresentano un punto di partenza per futuri sviluppi volti ad aumentare il livello di automazione del sistema, migliorarne la robustezza e favorirne l'impiego in ambito riabilitativo.

Bibliografia

[1] K. Li and P. A. Malhotra, “Spatial neglect,” *Pract. Neurol.*, vol. 15, no. 5, pp. 333–339, Oct. 2015, doi: 10.1136/PRACTNEUROL-2015-001115.

[2] M. Mancuso *et al.*, “From evidence to action: Italian recommendations for the diagnosis and treatment of spatial neglect in stroke patients,” *Neurological Sciences 2026 47:6*, vol. 47, no. 6, pp. 532-, May 2026, doi: 10.1007/S10072-026-09115-Z.

[3] J. Belger *et al.*, “Virtual reality eye movement training for neglect rehabilitation,” 2025, doi: 10.1016/j.ijchp.2025.100630.

[4] A. Martino Cinnera *et al.*, “Immersive Virtual Reality for Treatment of Unilateral Spatial Neglect via Eye-Tracking Biofeedback: RCT Protocol and Usability Testing,” *Brain Sciences 2024, Vol. 14, Page 283*, vol. 14, no. 3, p. 283, Mar. 2024, doi: 10.3390/BRAINSCI14030283.

[5] P. Petitet, J. X. O’Reilly, and J. O’Shea, “Towards a neuro-computational account of prism adaptation,” *Neuropsychologia*, vol. 115, pp. 188–203, Jul. 2018, doi: 10.1016/J.NEUROPSYCHOLOGIA.2017.12.021.

[6] A. Martino Cinnera *et al.*, “Immersive Virtual Reality for Treatment of Unilateral Spatial Neglect via Eye-Tracking Biofeedback: RCT Protocol and Usability Testing,” 2024, doi: 10.3390/brainsci14030283.

[7] J. Uimonen *et al.*, “Virtual reality tasks with eye tracking for mild spatial neglect assessment: a pilot study with acute stroke patients,” *Front. Psychol.*, vol. 15, p. 1319944, Jan. 2024, doi: 10.3389/FPSYG.2024.1319944/TEXT.

[8] M. Delazer, M. Sojer, P. Ellmerer, C. Boehme, and T. Benke, “Eye-tracking provides a sensitive measure of exploration deficits after acute right MCA stroke,” *Front. Neurol.*, vol. 9, no. JUN, p. 374184, Jun. 2018, doi: 10.3389/FNEUR.2018.00359/TEXT.

[9] D. Giansanti and C. Tisp, “The Social Robot in Rehabilitation and Assistance: What Is the Future?,” *Healthcare 2021, Vol. 9, Page 244*, vol. 9, no. 3, p. 244, Feb. 2021, doi: 10.3390/HEALTHCARE9030244.

[10] N. Nordin, S. Quan Xie, and B. Wünsche, “REVIEW Open Access Assessment of movement quality in robot-assisted upper limb rehabilitation after stroke: a review,” 2014. [Online]. Available: http://www.jneuroengrehab.com/content/11/1/137

[11] M. A. Roa *et al.*, “Mobile Manipulation Hackathon: Moving into real world applications,” 2021, doi: 10.1109/MRA.2021.3061951.

[12] “TIAGo | Mobile Manipulator Robot.” Accessed: Jun. 15, 2026. [Online]. Available: https://pal-robotics.com/robot/tiago/

[13] F. J. Naranjo-Campos, A. De Matías-Martínez, J. G. Victores, J. A. Gutiérrez Dueñas, A. Alcaide, and C. Balaguer, “Assistance in Picking Up and Delivering Objects for Individuals with Reduced Mobility Using the TIAGo Robot,” *Applied Sciences (Switzerland)*, vol. 14, no. 17, Sep. 2024, doi: 10.3390/app14177536.

[14] R. Soldi *et al.*, “A Modular Robotic Platform for Rehabilitation of Hemispatial Inattention,” 2025.

[15] B. M. V. Guerra *et al.*, “Fit4MedRob: A Next-Generation Platform for Intelligent Robotic Rehabilitation,” Institute of Electrical and Electronics Engineers (IEEE), Dec. 2025, pp. 578–585. doi: 10.1109/dsd67783.2025.00085.

[16] Tobii AB, “Tobii Pro Glasses 3 Brochure.” Accessed: Jun. 19, 2026. [Online]. Available: https://tidentech.com/wp-content/uploads/2025/03/TobIi_Pro_Glasses_3_leaflet_A4_RGB.pdf

[17] Tobii AB, *Tobii Pro Glasses 3 Product Description Head Unit Recording Unit Corrective Lenses Controller Application Product Description Tobii Pro Glasses 3*, Version 1.0.1. Tobii AB, 2020. [Online]. Available: www.tobiipro.com

[18] Tobii AB, *Tobii Pro Glasses 3 Developer Guide*, Version 1.7. Tobii AB, 2023.

[19] Tobii AB, *Tobii Pro Glasses 3 User Manual*, Version 1.20. Tobii AB, 2024.

[20] John Elvesjö; Mårten Skogo; Gunnar Elvers, “Method and Installation for Detecting and Following an Eye and the Gaze Direction Thereof,” US Patent 7,572,008 B2,” Aug. 11, 2009

[21] “How do Tobii eye trackers work?” Accessed: May 05, 2026. [Online]. Available: https://connect.tobii.com/s/article/How-do-Tobii-eye-trackers-work?language=en_US

[22] “How do eye trackers work? — A tech-savvy walk-through - Tobii.” Accessed: May 05, 2026. [Online]. Available:

https://www.tobii.com/resource-center/learn-articles/how-do-eye-trackers-work

[23] “Eye tracker calibration and validation.” Accessed: May 07, 2026. [Online]. Available: https://connect.tobii.com/s/article/eye-tracker-calibration?language=en_US

[24] Tobii AB, *Tobii Pro Lab User Manual*. Tobii AB, 2025.

[25] “Tobii Pro Lab Gaze Filter.” Accessed: May 06, 2026. [Online]. Available: https://connect.tobii.com/s/article/Gaze-Filter-functions-and-effects?language=en_US

[26] “The Tobii I-VT Fixation Filter Algorithm description,” 2012. [Online]. Available: www.tobii.com

[27] “Tobii Pro Lab Data export formats.” Accessed: May 06, 2026. [Online]. Available: https://connect.tobii.com/s/article/Tobii-Pro-Lab-Data-export-formats?language=en_US

[28] H. Drewes, K. Pfeuffer, and F. Alt, “Time- And space-efficient eye tracker calibration,” in *Eye Tracking Research and Applications Symposium (ETRA)*, Association for Computing Machinery, Jun. 2019. doi: 10.1145/3314111.3319818.

[29] V. Onkhar, D. Dodou, and J. C. F. de Winter, “Evaluating the Tobii Pro Glasses 2 and 3 in static and dynamic conditions,” *Behav. Res. Methods*, vol. 56, no. 5, pp. 4221–4238, Aug. 2024, doi: 10.3758/s13428-023-02173-7.

[30] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” *PeerJ Comput. Sci.*, vol. 7, pp. 1–24, 2021, doi: 10.7717/PEERJ-CS.623.

[31] “Annotation Best Practices | Building High-Performance YOLO Datasets | Ultralytics Academy.” Accessed: Jun. 13, 2026. [Online]. Available: https://academy.ultralytics.com/courses/dataset-readiness-for-yolo/annotation-best-practices

[32] “Roboflow: Computer vision tools for developers and enterprises.” Accessed: Jun. 13, 2026. [Online]. Available: https://roboflow.com/

[33] “How Much Training Data Do You Need for Computer Vision?” Accessed: Jun. 13, 2026. [Online]. Available: https://blog.roboflow.com/how-much-training-data/

[34] “Tips for Best YOLOv5 Training Results | Ultralytics Docs.” Accessed: May 14, 2026. [Online]. Available: https://docs.ultralytics.com/yolov5/tutorials/tips_for_best_training_results

[35] “Launch: Auto Label Images with Roboflow.” Accessed: May 13, 2026. [Online]. Available: https://blog.roboflow.com/launch-auto-label/

[36] “How Much Training Data Do You Need for Computer Vision?” Accessed: Jun. 13, 2026. [Online]. Available: https://blog.roboflow.com/how-much-training-data/

[37] “Preprocess Images | Roboflow Docs.” Accessed: Jun. 13, 2026. [Online]. Available: https://docs.roboflow.com/datasets/dataset-versions/image-preprocessing

[38] “Configuration | Ultralytics Docs.” Accessed: May 15, 2026. [Online]. Available: https://docs.ultralytics.com/usage/

[39] “Instance Segmentation Datasets Overview | Ultralytics Docs.” Accessed: May 13, 2026. [Online]. Available: https://docs.ultralytics.com/datasets/segment

[40] “Object Detection Datasets Overview | Ultralytics Docs.” Accessed: May 13, 2026. [Online]. Available: https://docs.ultralytics.com/datasets/detect#ultralytics-yolo-format

[41] “Detection model training with segmentation annotation · Issue #13166 · ultralytics/ultralytics.” Accessed: May 13, 2026. [Online]. Available: https://github.com/ultralytics/ultralytics/issues/13166

[42] “Home | Ultralytics Docs.” Accessed: Jun. 13, 2026. [Online]. Available: https://docs.ultralytics.com/

[43] “Explore Ultralytics YOLOv8 | Ultralytics Docs.” Accessed: Jun. 13, 2026. [Online]. Available: https://docs.ultralytics.com/models/yolov8

[44] F. Hermens, “Automatic object detection for behavioural research using YOLOv8,” *Behav. Res. Methods*, vol. 56, no. 7, pp. 7307–7330, Oct. 2024, doi: 10.3758/s13428-024-02420-5.

[45] “Install Ultralytics | Ultralytics.” Accessed: Jul. 06, 2026. [Online]. Available: https://docs.ultralytics.com/quickstart#can-i-install-ultralytics-yolo-using-conda

[46] “Model Training with Ultralytics YOLO | Ultralytics Docs.” Accessed: May 15, 2026. [Online]. Available: https://docs.ultralytics.com/modes/train

[47] “Data Augmentation using Ultralytics YOLO | Ultralytics Docs.” Accessed: May 14, 2026. [Online]. Available: https://docs.ultralytics.com/guides/yolo-data-augmentation#custom-albumentations-transforms-augmentations

[48] “Dynamic On-the-Fly Augmentation.” Accessed: Jul. 07, 2026. [Online]. Available: https://www.emergentmind.com/topics/on-the-fly-augmentation-capability

[49] “Performance Metrics Deep Dive | Ultralytics Docs.” Accessed: May 15, 2026. [Online]. Available: https://docs.ultralytics.com/guides/yolo-performance-metrics#introduction

[50] “Model Validation with Ultralytics YOLO | Ultralytics Docs.” Accessed: Jun. 13, 2026. [Online]. Available: https://docs.ultralytics.com/modes/val#example-validation-with-arguments

[51] “Model Prediction with Ultralytics YOLO | Ultralytics Docs.” Accessed: Jun. 13, 2026. [Online]. Available: https://docs.ultralytics.com/modes/predict

[52] “Linear Interpolation Formula - Derivation, Formulas, Examples.” Accessed: May 20, 2026. [Online]. Available: https://www.cuemath.com/linear-interpolation-formula/

[53] “How does blinking affect eye tracking?” Accessed: May 20, 2026. [Online]. Available: https://connect.tobii.com/s/article/how-does-blinking-affect-eye-tracking?language=en_US

[54] R. Hershman, A. Henik, and N. Cohen, “A novel blink detection method based on pupillometry noise,” *Behav. Res. Methods*, vol. 50, no. 1, pp. 107–114, Feb. 2018, doi: 10.3758/s13428-017-1008-1.

[55] J. W. Grootjen, H. Weingärtner, and S. Mayer, “Highlight-ing the Challenges of Blinks in Eye Tracking for Interactive Systems,” *2023 Symposium on Eye Tracking Research and Applications (ETRA ’23), May 30-June 2, 2023, Tubingen, Germany*, vol. 1, 2023, doi: 10.1145/3588015.3589202.