Alma Mater Studiorum – Università di Bologna – Campus di Cesena

Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione

"Guglielmo Marconi"

Corso di Laurea in Ingegneria Biomedica

TITOLO DELL'ELABORATO

ViFoSe: modulo per la segmentazione automatica del foreground da video stabilizzati e strumenti per la correzione manuale.

Elaborato in

Fondamenti di Ingegneria dei Tessuti Biologici LM

Relatore

Prof. Emanuele Domenico Giordano

Presentata da

Michele Sarneri

Correlatori

Prof. Filippo Piccinini

Prof. Gastone Castellani

Dott. Nicola Normanno

Anno Accademico 2024 / 2025

"*La vita è ciò che accade mentre siamo impegnati a fare altri progetti.*"

John Lennon

Parole Chiave

Computer Vision

Image Processing

Video Analysis

Foreground Detection

Morphological Operators

Abstract (versione Italiana)

Nel campo della microscopia ed in molti altri settori, l'analisi di video acquisiti con telecamere è fondamentale per studiare il comportamento degli oggetti in movimento (primo piano) rispetto a diversi scenari (sfondo). Tra le diverse tecniche proposte in letteratura, i modelli di deep learning costituiscono uno degli approcci più efficaci per isolare gli elementi dinamici nelle immagini. Questo lavoro esplora algoritmi e strumenti per la rimozione dello sfondo in video con sfondi fissi e variabili, concentrandosi anche su procedure di post-elaborazione. L'algoritmo sviluppato utilizza un modello di intelligenza artificiale chiamato *Segment Anything Model* (*SAM*) in grado di identificare in modo semi automatico gli elementi di interesse da segmentare all'interno di un filmato. Algoritmi successivi di processamento vengono aggiunti per perfezionare la segmentazione attraverso l'utilizzo di filtri, inviluppo convesso e operazioni morfologiche. Per risolvere problematiche come rumore, effetti di camuffamento e variazioni dinamiche, sono integrati filtri e tecniche di riempimento. L'algoritmo è testato e applicato in numerose categorie di filmati, in modo particolare microscopia, biologia e videosorveglianza. Il risultato finale corrisponde ad un video in cui lo sfondo è completamente rimosso, lasciando visibile solo il primo piano. Lo studio mostra l'efficacia delle tecniche di post-elaborazione per una segmentazione video accurata e robusta. L'algoritmo complessivo è stato integrato all'interno del software *ViFoSe*, disponibile tramite link nell'apposita pagina *GitHub*, all'interno della quale sono riportate in modo chiaro tutte le specifiche riguardo l'utilizzo del programma.

Abstract (English version)

In the field of microscopy and in many other domains, the analysis of videos acquired by cameras is essential for studying the behavior of moving objects (foreground) with respect to different scenarios (background). Among the various techniques proposed in the literature, deep learning models represent one of the most effective approaches for isolating dynamic elements in images. This work investigates algorithms and tools for background removal in videos with both static and dynamic backgrounds, with a particular focus on post-processing procedures. The developed algorithm employs an artificial intelligence model known as the *Segment Anything Model* (*SAM*), which is capable of semi-automatically identifying the elements of interest to be segmented within a video sequence. Subsequent algorithms are applied to refine the segmentation, including the use of filtering methods, convex hulls, and morphological operations. To address issues such as noise, camouflage effects, and dynamic variations, filters, and filling techniques are integrated. The algorithm is tested and applied to several categories of video data, with particular emphasis on microscopy, biology, and video surveillance. The final output consists of a video in which the background is completely removed, leaving only the foreground visible. The study demonstrates the effectiveness of post-processing techniques in achieving accurate and robust video segmentation. The complete algorithm has been integrated into the *ViFoSe* software, which is available via a link on the corresponding *GitHub* page, where detailed specifications and usage instructions are clearly provided.

Indice

Introduzione

Questo progetto di Tesi è stato sviluppato all'interno del gruppo di ricerca "*Microscopy & Artificial Intelligence*" (MiAI), composto da ricercatori e professori dell'Università di Bologna (UniBo) e dell'*Istituto Romagnolo per lo Studio dei Tumori "Dino Amadori"* (IRST) di Meldola (FC). Il gruppo MiAI si propone di coordinare professionisti e risorse provenienti dai settori dell'informatica, dell'ingegneria dell'informazione, della fisica, nonché dalle discipline biomediche, della biologia, della chimica e della medicina. In particolare, il progetto di Tesi intitolato "*ViFoSe: modulo per la segmentazione automatica del foreground da video stabilizzati e strumenti per la correzione manuale*" nasce da una specifica necessità sorta nel contesto del gruppo di ricerca di estrarre da video acquisiti con microscopi gli oggetti di interesse eliminando informazioni relative allo sfondo sottostante. L'obiettivo di questo progetto è fornire a medici e biologi un tool estremamente facile da utilizzare sviluppato in forma open source affinché la comunità scientifica possa in futuro estendere le funzionalità integrando nuovi algoritmi.

Il lavoro di tesi svolto rappresenta l'insieme di una serie di progetti, che hanno portato alla creazione di una delle principali funzionalità per il software finale *ViFoSe*, tale da permettere l'estrazione del primo piano (foreground, FG) da diversi formati di video.

I video sono in genere filmati più brevi che mirano a catturare momenti, eventi o idee un po' più nello specifico e, per prendere chiarezza sul lavoro svolto è necessario ricordare alcuni termini specifici, come il primo piano e lo sfondo, i quali sono legati alla relazione spaziale all'interno di un fotogramma in un file video o cinematografico. Il primo piano è l'area che considera il soggetto/i principale/i, mentre lo sfondo (background, BG) è l'area collocata dietro il soggetto/i. A livello pratico, il primo piano racchiude solitamente gli elementi più rilevanti che catturano l'attenzione dello spettatore come gli 'oggetti' in movimento, nel nostro caso identificati come oggetti d'interesse (Objects of Interest, OOIs), mentre le informazioni statiche definiscono lo sfondo, il quale fornisce il contesto o in generale l'ambientazione [1]. Solitamente queste tipologie di video sono acquisite per scopi specifici, in particolare nel contesto scientifico possono essere realizzate per condurre analisi riguardo cellule o microrganismi in movimento. Nel lavoro sono stati analizzati filmati con sfondi costanti (statici) e variabili (dinamici), con uno o più soggetti in movimento in primo piano.

La segmentazione video permette di separare gli oggetti in primo piano dallo sfondo in un'immagine o in un video, così da potersi concentrare sugli elementi più rilevanti della scena. Essa costituisce un

passaggio fondamentale nel cotesto della Computer Vision e nelle applicazioni di Computer Graphics, come il rilevamento degli oggetti, il tracciamento e l'analisi del movimento [2]. Esistono diverse tecniche per la segmentazione primo piano-sfondo, ognuna con i propri punti di forza e limiti. Tale procedura varia a seconda della tecnica utilizzata e dai requisiti specifici dell'applicazione; per esempio le applicazioni moderne che lavorano in tempo reale richiedono algoritmi più veloci, mentre nel contesto delle analisi offline è preferibile una maggiore accuratezza.

Nel progetto svolto, viene specificatamente utilizzato il modello di deep learning pre-addestrato *Segment Anything Model* (*SAM*), basato su un approccio generativo, tale da guidare con accuratezza il processo di segmentazione, con l'obiettivo di isolare gli OOIs dallo sfondo mantenendo per ogni fotogramma solamente le informazioni relative agli elementi in movimento.

In questa Tesi sono proposti i principali concetti e dettagli riguardo all'Analisi Video, analizzando in profondità i dettagli di *SAM*, esplorando le caratteristiche primarie, valutando le sue prestazioni in diversi dataset acquisiti e confrontandolo con altri metodi di segmentazione. L'obiettivo principale è stato implementare e validare il modello di segmentazione alla base del software *ViFoSe*, sviluppato per consentire la segmentazione del foreground in qualunque condizione di acquisizione video, individuando criticità e potenziali direzioni future.

Stato dell'Arte

In letteratura, le tecniche di segmentazione possono essere suddivise in tre principali categorie: i metodi basati su *modelli* [3], quelli basati sul *contesto* [4] e quelli basati su deep learning [5]. I primi fanno affidamento a conoscenze pregresse dello sfondo, come le proprietà statistiche o l'aspetto, mentre i secondi non richiedono informazioni preliminari e sfruttano le proprietà intrinseche dell'immagine o della sequenza video. Infine, i metodi basati sul deep learning si affidano a modelli guidati dai dati, addestrati su grandi dataset, per eseguire la segmentazione del foreground sulla base di rappresentazioni visive apprese, piuttosto che su una modellazione esplicita del background.

Tra i metodi basati su modelli, uno degli approcci più diffusi è la *sottrazione dello sfondo* [6], che consiste nel confrontare ogni fotogramma con un modello di sfondo precedentemente stimato al fine di individuare le regioni in movimento. A questo si affiancano tecniche più sofisticate, come i modelli di *mescolanza gaussiana* (GMM) [7] e la *stima della densità del kernel* (KDE) [8], che permettono di modellare la distribuzione dei pixel di sfondo e primo piano. Sebbene tali approcci risultino più robusti alle variazioni di illuminazione o a lievi cambiamenti della scena, essi presentano un costo computazionale maggiore e richiedono una fase di inizializzazione accurata.

I metodi basati sul contesto includono anche tecniche come *Graph Cuts* [9], *Active Contours* [10] e *Level Sets* [11]. I primi due si basano rispettivamente sulla minimizzazione dell'energia e sulla valutazione delle curve per separare il primo piano dallo sfondo. Il terzo, invece, sfrutta equazioni differenziali per evolvere la rappresentazione del confine tra primo piano e sfondo. Questi metodi risultano più flessibili, in quanto non necessitano di conoscenze pregresse, ma sono spesso sensibili al rumore e dipendono dalla qualità dell'inizializzazione.

I metodi fondati su deep learning per la segmentazione di primo piano e sfondo hanno ricevuto una notevole attenzione grazie ai progressi delle reti neurali convoluzionali (*CNN*) e nelle architetture correlate, che apprendono caratteristiche visive direttamente dai dati anziché basarsi su modelli progettati manualmente o su rappresentazioni esplicite dello sfondo. Nel contesto della segmentazione semantica, le reti encoder–decoder come *U-Net* [12] e le *Fully Convolutional Networks* (FCNs) [13] sono state ampiamente adottate per produrre maschere pixel-wise apprendendo da dataset annotati, ottenendo prestazioni elevate in condizioni di imaging eterogenee sia per immagini sia per video

Nell'analisi video in post-produzione, si possono identificare diverse categorie di video:

a) Sfondo fisso, telecamera fissa, singolo oggetto sempre visibile:
   La camera e lo sfondo sono statici, e l'oggetto di interesse è sempre visualizzabile. Esempi:
   - Filmati di videosorveglianza;
   - Video didattici semplici.

b) Sfondo dinamico, telecamera fissa, singolo oggetto sempre visibile:
   Lo sfondo cambia nel tempo, mentre l'oggetto rimane visibile. Esempi:
   - Video di una spiaggia con onde in movimento;
   - Scene urbane con variazioni di illuminazione;
   - Interni con caminetto acceso.

c) Sfondo fisso, telecamera in movimento, singolo oggetto sempre visibile:
   La telecamera si sposta o oscilla mentre lo sfondo rimane invariato. È necessario allineare i fotogrammi prima dell'analisi. Esempi:
   - Riprese con drone su un campo statico;
   - Telecamera che segue una persona in una stanza;
   - Video girato a mano con leggere oscillazioni.

   Uno dei tipici algoritmi che viene utilizzato per questo tipo di acquisizioni è la *Correlazione di Fase* [14].

d) Sfondo dinamico, telecamera in movimento, singolo oggetto sempre visibile:
   Sia la camera che lo sfondo sono dinamici, introducendo problemi come vibrazioni, sfocature da movimento e variazioni di sfondo. Esempi:
   - Riprese sportive con pubblico in movimento;
   - Video naturalistici con telecamera in movimento.

e) Video con più oggetti:
   La presenza di più oggetti rende più complessa la segmentazione e il tracciamento. Esempi:
   - Filmati del traffico con più veicoli;
   - Performance di danza di gruppo.

f) Video con oggetti non sempre visibili:
   Gli oggetti possono scomparire temporaneamente per occlusioni o cambiamenti di luce. Richiedono tecniche di tracciamento predittivo e ri-identificazione degli oggetti. Esempi:
   - Tracciamento di pedoni in un'area urbana;
   - Video sportivi con cambi di inquadratura.

Negli ultimi anni, l'evoluzione delle reti neurali profonde ha portato allo sviluppo di approcci data-driven, in grado di superare molte delle limitazioni dei metodi tradizionali. È in questo contesto che si colloca il *SAM* [15], un modello di segmentazione universale basato su deep learning, progettato per segmentare qualsiasi oggetto presente in un'immagine senza la necessità di un addestramento specifico sul dominio di interesse. *SAM* sfrutta un'architettura transformer e un vasto dataset di addestramento, consentendo una segmentazione accurata e generalizzabile anche in scenari complessi, con sfondi statici o dinamici e con uno o più oggetti.

A differenza degli approcci classici, *SAM* non richiede la costruzione di un modello di sfondo né assunzioni sulle caratteristiche della scena; esso può essere guidato tramite prompt, rendendolo particolarmente adatto a contesti di post-produzione video, dove è possibile integrare l'intervento umano per ottenere segmentazioni di alta qualità. Tali modelli possono essere estesi in modo efficace all'analisi di sequenze video, applicando la segmentazione frame-by-frame o introducendo meccanismi di propagazione temporale delle maschere.

Alla luce di queste considerazioni, il lavoro svolto si concentra sull'impiego di *SAM* per l'analisi di video con telecamera fissa e sfondo statico o leggermente dinamico, in presenza di uno o più soggetti in movimento continuo e discontinuo. L'obiettivo è valutare l'efficacia di questo approccio, evidenziandone i vantaggi rispetto alle tecniche tradizionali e analizzandone le principali criticità nel contesto dell'elaborazione video.

3 Analisi Video

Prima di entrare nel dettaglio del lavoro svolto è opportuno chiarire il significato del termine ‘video d’immagini’. Un video può essere definito come una sequenza temporale di immagini statiche, denominate fotogrammi, visualizzate in rapida successione secondo uno specifico frame rate (ad esempio 24, 30 o 60 fotogrammi al secondo). L’estrazione e l’analisi dei singoli fotogrammi permettono di esaminare il contenuto del filmato complessivo in modo più accurato e controllato, consentendo un'elaborazione più dettagliata e approfondita. In questo contesto, il lavoro si è focalizzato sulla scomposizione dei video originali nei rispettivi fotogrammi [16], al fine di condurre un’analisi dettagliata frame by frame.

3.1 Struttura di un video d’immagini

Un file video è composto da una sequenza ordinata di fotogrammi, ciascuno dei quali rappresenta una singola immagine statica codificata secondo uno specifico formato di compressione. Ogni fotogramma cattura lo stato della scena in un preciso istante temporale, fornendo una rappresentazione dell’evoluzione temporale del contenuto visivo. La riproduzione consecutiva dei fotogrammi a una determinata frequenza, definita ‘frame rate’, genera la percezione di continuità e movimento tipica dei video. Ai fini di un’analisi accurata, in particolare nell’ambito dell’elaborazione digitale delle immagini e della visione artificiale, risulta vantaggioso accedere ai fotogrammi in modo sequenziale e trattarli come singole entità. Questo approccio consente di applicare algoritmi di segmentazione a livello di singola immagine, facilitando l’estrazione di informazioni visive e permettendo un controllo preciso sulle operazioni effettuate su ciascun istante della sequenza.

Estrazione dei fotogrammi

1) **Caricamento del video:**
   La prima fase del processo consiste nel caricamento del file video utilizzando strumenti (software di elaborazione) o librerie di elaborazione in grado di supportare i formati più comuni (es. MP4, AVI). Il video, memorizzato in forma ‘compressa’, viene interpretato come un flusso di dati che consente l’accesso sequenziale ai fotogrammi, rendendo possibile la loro decodifica e successiva elaborazione.

2) **Calcolo del Frame Rate**:

   Un passaggio importante per la corretta gestione della sequenza video consiste nel determinare il *Frame Rate* (Frame Per Second, FPS), che indica il numero di fotogrammi visualizzati in un secondo. Tale valore può essere calcolato dividendo il numero totale di frame per la durata complessiva del video espressa in secondi. Nel progetto svolto, i filmati presi come riferimento presentano principalmente un frame rate con un valore pari a 30 FPS.

3) **Lettura frame by frame**:

   I video digitali sono generalmente composti da diverse tipologie di frame, utilizzate per ottimizzare la compressione:

   - *I-Frame* (Intra-Coded Frames): fotogrammi completi indipendenti dagli altri;
   - *P-Frame* (Predicted Frames): fotogrammi che contengono solo le variazioni rispetto a frame precedenti;
   - *B-Frame* (Bidirectional Frames): fotogrammi che dipendono sia dai frame precedenti sia da quelli successivi;

   Per poter accedere al contenuto visivo di ciascun fotogramma, è necessario eseguire una prima fase di decodifica, al termine della quale ogni frame viene rappresentato in memoria come una matrice di pixel, generalmente organizzata secondo i canali di colore.

4) **Salvataggio e Accessibilità dei fotogrammi**:

   Al termine del processo di estrazione, i fotogrammi vengono salvati come singoli file immagine all'interno di una directory di output, utilizzando formati standard (ad esempio jpeg, png, bmp, tiff). Nel presente lavoro, è stato scelto il formato PNG, in quanto consente una compressione senza perdita di informazione, preservando la qualità originale dei frame. Ogni immagine viene inoltre nominata in modo sequenziale (ad esempio frame_00000.png, frame_00001.png, etc…), garantendo un ordine nel collocamento dei file all'interno della directory e, facilitando il riferimento, l'accesso e le successive fasi di analisi frame by frame.

Esempio di rappresentazione dei fotogrammi

Dato un video con risoluzione spaziale pari a $W \times H$ (larghezza per altezza, espressa in pixel) e composto da un totale di N fotogrammi, è possibile rappresentare l'insieme dei fotogrammi come una sequenza ordinata:

$$F=\{f_1, f_2, f_3, \ldots, f_N\}$$

dove f_i rappresenta l' i-esimo fotogramma della sequenza video. Ciascun fotogramma f_i rappresenta lo stato della scena in un determinato istante e può essere descritto da una struttura dati bidimensionale o tridimensionale, a seconda della tipologia di contenuto visivo.

- Nel caso di un video in scala di grigi, ciascun fotogramma f_i è descritto da una matrice bidimensionale di dimensioni $W \times H$, in cui ogni elemento rappresenta il valore di intensità luminosa associato a un singolo pixel:

$$f_i \in R^{W \times H}$$

- Per un video a colori, tipicamente rappresentato secondo il modello RGB, ciascun fotogramma f_i è invece descritto da un tensore tridimensionale di dimensioni $W \times H \times 3$, dove per ogni pixel sono presenti tre valori che rappresentano rispettivamente i canali rosso (R), verde (G) e blu (B):

$$f_i \in R^{W \times H \times 3}$$

L'intero video può quindi essere visto come un tensore di dimensioni $W \times H \times C \times N$, dove C indica il numero di canali (1 nel caso della scala di grigi, 3 nel caso RGB) e N il numero totale di fotogrammi che compongono la sequenza.

Calcolo del numero totale di fotogrammi

Il numero complessivo di fotogrammi N contenuto in un video dipende dalla frequenza di acquisizione dei fotogrammi (frame rate) e dalla durata complessiva del video. Tale relazione può essere espressa come:

$$N = F \times T$$

dove:

- F rappresenta la frequenza dei fotogrammi, misurata in fotogrammi per secondo (fps);
- T indica la durata totale del video, espressa in secondi.

Da questa relazione emerge che un valore più elevato di F consente di ottenere una rappresentazione temporale più fluida e dettagliata del movimento, mentre una riduzione del frame rate comporta una

minore continuità temporale e un maggiore contrasto tra i fotogrammi consecutivi. Tale aspetto risulta particolarmente rilevante nelle applicazioni di analisi video e visione artificiale, dove la risoluzione temporale influisce direttamente sulla qualità dei risultati ottenuti.

3.2 Tipologie di formati video

È opportuno chiarire il significato di 'formato video', termine che indica l'insieme di specifiche tecniche che definiscono le modalità con cui i dati video vengono compressi, memorizzati e riprodotti. Un formato video stabilisce quindi la struttura interna del file e le regole secondo cui i flussi di video, audio ed eventuali informazioni aggiuntive possono essere interpretati da dispositivi hardware e software differenti.

In questo scenario, si utilizza solitamente il termine 'Container' per riferirsi al file che racchiude uno o più flussi di dati, quali audio, video, sottotitoli e metadati. Esistono numerose tipologie di container, ciascuna caratterizzata da specifiche proprietà in termini di compatibilità, qualità e compressione. Tra i più diffusi si possono distinguere:

- Formato **AVI (Audio Video Interleave)**:
    - Sviluppato da Microsoft nel 1992;
    - Ampliamente compatibile con Windows, VLC e diversi software di editing;
    - Consente un'alta qualità video, ma genera file di grandi dimensioni, per cui la compressione risulta limitata.

- Formato **MP4 (MPEG-4 Part 14)**:
    - Sviluppato dal gruppo MPEG nel 2001;
    - Elevata compatibilità con un'ampia gamma di dispositivi e piattaforme (PC, Mac, smartphone, TV e web);
    - Offre un buon compromesso tra qualità e dimensione del file, sebbene compressioni elevate possano comportare una perdita di dettaglio.

- Formato **MOV (QuickTime File Format)**:
    - Sviluppato da Apple nel 1991;

    - Ottimizzato per l'ambiente MacOS, QuickTime e software di editing professionali;
    - Garantisce una qualità video molto elevata, a fronte di dimensioni dei file maggiori e di una compatibilità più limitata con sistemi non Apple.

- Formato **MKV (Matroska Video)**:
    - Formato open-source introdotto nei primi anni 2000;
    - Supporta molteplici flussi audio, video, sottotitoli e metadati all'interno di un unico file;
    - È caratterizzato da elevata flessibilità e qualità, con ampia compatibilità in player come VLC, sebbene non sempre supportato da tutti i dispositivi.

- Formato **WebM**:
    a) Sviluppato da Google e progettato principalmente per la distribuzione di contenuti video sul web;
    b) È basato su codec open-source, come VP8/VP9 e Opus;
    c) Risulta ottimizzato per lo streaming online, mantenendo un buon compromesso tra qualità visiva e riduzione della dimensione dei file.

Per fornire un riferimento concreto alle specifiche tecniche descritte, nel seguito vengono mostrati alcuni fotogrammi rappresentativi di un video acquisito durante l'esperienza sperimentale, selezionati rispettivamente all'inizio, a metà e al termine della sequenza. L'analisi di tali fotogrammi consente di osservare l'evoluzione temporale dei soggetti dinamici presenti nella scena, permettendo di studiare eventuali variazioni nelle traiettorie, nei movimenti e nelle caratteristiche visive complessive del contenuto video.

Figura 1: esempio del primo frame estratto dal video originale.

Figura 2: esempio di frame estratto a circa metà del video originale.

Figura 3: esempio dell'ultimo frame estratto dal video originale.

3.3 Foreground

Nel contesto dell'analisi di un video d'immagini, con il termine 'primo piano' (foreground) si indica l'insieme degli elementi della scena che risultano maggiormente rilevanti dal punto di vista percettivo e informativo. Tali elementi coincidono spesso con i soggetti principali del video e rappresentano il centro dell'attenzione dell'osservatore. Il foreground si distingue dallo sfondo (background) e dal piano intermedio (middleground), i quali contribuiscono alla strutturazione spaziale della scena e alla percezione della profondità, ma rivestono un ruolo secondario rispetto agli elementi in primo piano.

Caratteristiche del Foreground

Il primo piano presenta generalmente alcune caratteristiche distintive:

- **Maggiore livello di dettaglio**: essendo spesso più vicino all'osservatore o alla telecamera, il foreground è caratterizzato da una maggiore risoluzione spaziale, con contorni più definiti e una maggiore quantità di dettagli visivi rispetto gli altri piani della scena.

- **Illuminazione e contrasto**: gli elementi in primo piano tendono a presentare valori di illuminazione e contrasto maggiori, tali da renderli distinguibili dallo sfondo, facilitando la percezione e l'analisi visiva.
- **Rilevanza semantica**: il foreground include tipicamente il soggetto o i soggetti di interesse, elementi che trasmettono l'informazione principale del contenuto video e che risultano rilevanti per l'osservatore o per l'algoritmo di analisi.
- **Dinamica temporale**: nei video, il primo piano è frequentemente associato a variazioni temporali significative, come movimenti, deformazioni o cambiamenti di posizione all'interno della scena.

Tecniche di composizione del Foreground

Nella costruzione di immagini e video, diverse tecniche compositive vengono adottate per enfatizzare il foreground e guidare l'attenzione dell'osservatore:

1) **Regola dei terzi**: l'inquadratura viene suddivisa in una griglia 3 × 3, dove gli elementi di maggiore importanza sono generalmente collocati lungo le linee guida o lungo i punti di intersezione. In genere, posizionare un elemento del primo piano lungo una delle linee verticali o orizzontali aiuta a bilanciare la scena; una disposizione in corrispondenza di un punto di intersezione invece, permette di acquisire maggiore importanza nella composizione. Predisporre il foreground secondo queste modalità, contribuisce a creare un equilibrio tra primo piano e sfondo, evitando che un elemento sia predominante sull'altro, con una distribuzione visiva eccessivamente centrata o sbilanciata.

2) **Utilizzo della prospettiva**: in generale la disposizione spaziale degli elementi in primo piano rispetto a quelli sullo sfondo, rafforza la percezione tridimensionale della scena. Fenomeni come la sovrapposizione spaziale degli oggetti (ad esempio un elemento in primo piano che copre parzialmente un altro in secondo piano), la variazione apparente delle dimensioni e le linee prospettiche contribuiscono a stabilire una gerarchia spaziale chiara, migliorando la comprensione delle distanze relative tra gli elementi.

3) **Utilizzo della profondità di campo (Depth of Field , DoF)**: la profondità di campo determina quali regioni della scena risultano a fuoco e quali appaiono sfocate. Il controllo della *DoF* consente di enfatizzare il foreground tramite diverse strategie:
    - Foreground nitido e background sfocato, per isolare ed enfatizzare il soggetto/i principale/i;
    - Utilizzo di elementi sfocati in primo piano come cornice visiva del soggetto (ad esempio oggetti vicino all'obiettivo);
    - Mantenimento di tutti i piani della scena a fuoco, al fine di avere una lettura chiara dei diversi elementi che compongono la scena.

A supporto delle considerazioni esposte, nel seguito viene presentato un fotogramma rappresentativo dell'esperienza sperimentale, estratto approssimativamente poco dopo l'inizio della sequenza video. Nell'immagine sono evidenziati i soggetti principali in movimento, i quali costituiscono il foreground del filmato e rappresentano le regioni di interesse per le successive fasi di analisi e segmentazione.

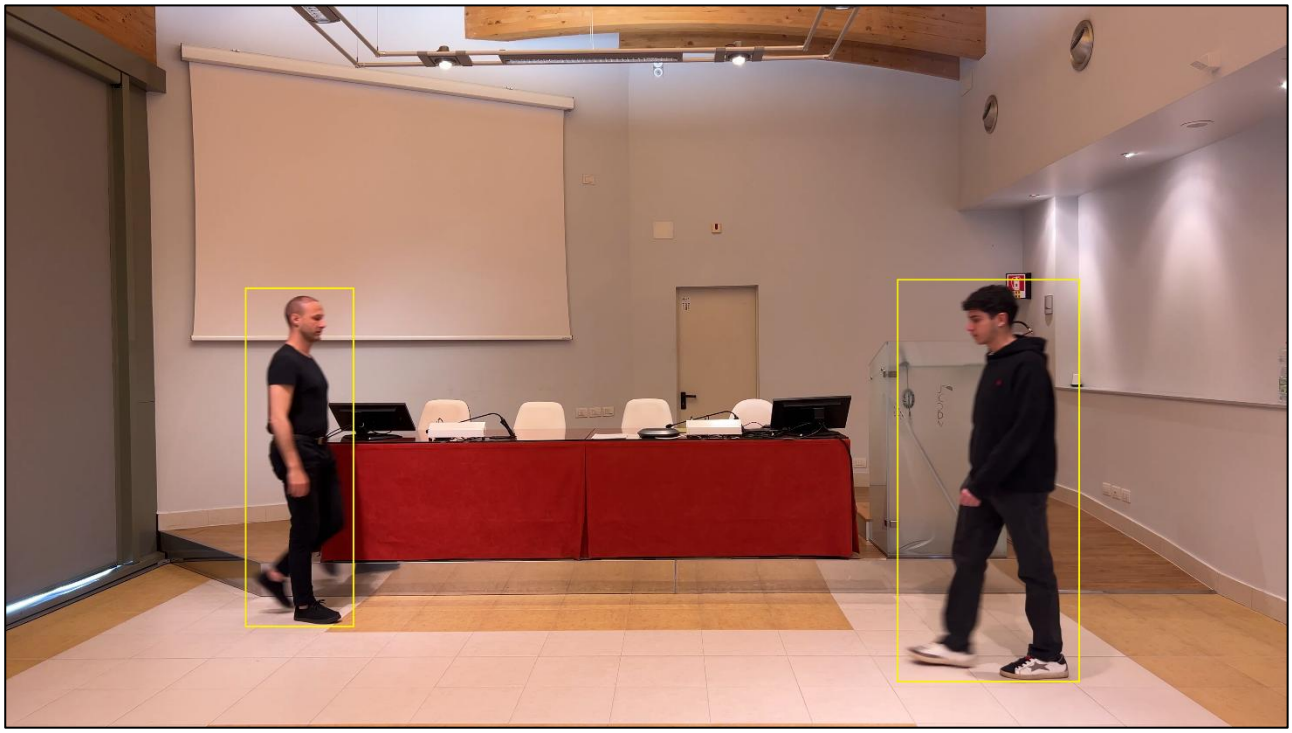

Figura 4: esempio di un frame estratto poco dopo l'inizio del video originale.

Foreground nel campo della Microscopia

Nel contesto della microscopia, il primo piano che identifica l'insieme degli elementi di interesse biologico presenti nell'immagine, è tipicamente rappresentato da cellule, tessuti, batteri, strutture subcellulari o particelle microscopiche, che costituiscono l'oggetto principale dell'osservazione e dell'analisi. Tali elementi si distinguono dallo sfondo, che può includere regioni non informative, rumore di acquisizione, artefatti ottici, oppure il mezzo in cui il campione è immerso, come soluzioni saline o substrati di supporto.

L'estrazione del foreground riveste un ruolo fondamentale nell'elaborazione delle immagini microscopiche, in quanto consente di isolare le strutture rilevanti dallo sfondo e di migliorare l'accuratezza delle successive analisi qualitative e quantitative. Un'adeguata separazione tra primo piano e sfondo permette infatti di ridurre l'influenza del rumore e degli artefatti, aumentando l'affidabilità delle misurazioni e delle interpretazioni biologiche.

L'individuazione ed estrazione del foreground in microscopia trova applicazioni in numerosi ambiti, tra cui:

- l'analisi e la misura delle strutture biologiche, come il conteggio cellulare, la valutazione della morfologia e il tracciamento del movimento di cellule o particelle;
- il supporto alla diagnosi medica mediante sistemi automatizzati, ad esempio per il riconoscimento di cellule tumorali, l'analisi di infezioni batteriche o l'identificazione di anomalie ematologiche;
- l'individuazione di regioni di interesse marcate tramite tecniche di fluorescenza, utili per evidenziare specifiche componenti cellulari o molecolari.

Per l'estrazione del foreground in ambito microscopico sono state sviluppate diverse strategie, spesso combinate tra loro per migliorare la robustezza dei risultati:

- **Filtraggio dell'immagine**: questa fase preliminare mira a perfezionare la qualità visiva dell'immagine e a semplificare le successive operazioni di segmentazione. Vengono impiegati filtri passa-basso per ridurre il rumore e dettagli indesiderati, oppure filtri passa-alto per enfatizzare contorni e dettagli fini, aumentando la nitidezza delle strutture d'interesse.
- **Segmentazione delle immagini**: in questo processo l'immagine è divisa in più parti per separare gli oggetti d'interesse dallo sfondo; possono essere utilizzate delle soglie (fisse o

adattive) [17], raggruppamenti e suddivisione dei pixel oppure metodi basati sull'individuazione dei bordi tramite variazioni d'intensità.

- **Tecniche di miglioramento del contrasto**: queste tecniche mirano a rendere più evidenti le differenze tra foreground e background attraverso la redistribuzione dei livelli di intensità. Possono includere operazioni globali o locali di 'contrast enhancement', nonché trasformazioni morfologiche volte a selezionare e rafforzare specifiche regioni del primo piano.

A titolo esemplificativo, nel seguito è mostrato un fotogramma rappresentativo di un' acquisizione in microscopia, in cui sono evidenziati alcuni dei soggetti principali, come microrganismi, che costituiscono il foreground della sequenza video. Tale rappresentazione consente di visualizzare in modo chiaro la separazione tra gli elementi di interesse e lo sfondo, facilitando le successive fasi di analisi.

Figura 5: esempio di un frame estratto a circa metà del video originale.

3.4 Background

Il background nei video d'immagini rappresenta un elemento concettuale e operativo di primaria importanza, in quanto influisce direttamente sulla qualità dell'analisi delle scene e sull'efficacia dei processi di estrazione delle informazioni rilevanti. Esso comprende l'insieme degli elementi presenti nella scena che non costituiscono il soggetto/i principale/i dell'osservazione, ovvero il foreground. A seconda del contesto applicativo e delle condizioni di acquisizione, lo sfondo può risultare statico, dinamico o soggetto a variazioni temporali e spaziali, introducendo livelli di complessità.

Dal punto di vista dell'elaborazione delle immagini, una corretta modellazione dello sfondo è essenziale per applicazioni quali la segmentazione, il rilevamento del movimento e il tracciamento degli oggetti, poiché una separazione inaccurata tra background e foreground può compromettere significativamente i risultati ottenuti.

Natura del Background (Statico e Dinamico)

Una prima distinzione fondamentale riguarda la natura temporale dello sfondo:

- **Background Statico**: è caratterizzato da elementi che rimangono invariati nel tempo. Questo scenario rappresenta una condizione ideale per molte applicazioni di *background subtraction*, in quanto lo sfondo può essere stimato con facilità e sottratto alla scena per isolare il foreground. Tali condizioni sono tipiche di ambienti controllati, come riprese in interni con telecamera fissa.
- **Background Dinamico**: in questo caso lo sfondo è soggetto a variazioni temporali e spaziali, dovute a cambiamenti di illuminazione, riflessi, presenza di oggetti in movimento, vibrazioni o spostamenti della telecamera. La distinzione tra foreground e background diventa quindi più complessa e richiede approcci più robusti. Questo tipo di scenario è comune in ambienti esterni, nella microscopia digitale e in applicazioni di robotica e videosorveglianza.

Per fornire maggiore chiarezza, di seguito è rappresentata un'immagine associata alla ricostruzione di un background statico, associato ad un filmato di videosorveglianza acquisito durante l'esperienza sperimentale. Nell'immagine non sono presenti parti in movimento, di conseguenza è possibile individuare solamente gli elementi statici che compongono la scena.

Figura 6: esempio di ricostruzione del background statico in un filmato di videosorveglianza.

Metodi di Estrazione del Background

L'estrazione e la modellazione del background costituiscono un passaggio fondamentale nei processi di separazione tra primo piano e sfondo. In letteratura sono stati proposti numerosi approcci, tra cui:

- **Modelli basati su medie e mediane**, dove lo sfondo viene stimato calcolando la media o la mediana dei valori dei pixel su una sequenza di fotogrammi, assumendo che gli oggetti in movimento occupino solo una parte limitata della scena. Questi metodi sono efficaci in presenza di background statici, ma risultano poco adatti a scenari dinamici.
- **Gaussian Mixture Models (GMM)**, per cui lo sfondo è modellato come una combinazione di distribuzioni gaussiane che consentono di rappresentare variazioni moderate nel tempo. Questo approccio è ampiamente utilizzato per la sua capacità di adattarsi a cambiamenti graduali della scena.
- **Trasformate di Fourier e Filtri Spaziali**, corrispondenti a tecniche che sfruttano la separazione delle componenti in frequenza per distinguere le variazioni rapide associate al foreground dalle componenti più lente, tipicamente riconducibili allo sfondo.

- **Kernel Density Estimation (KDE)**, relativa ad una tecnica non parametrica, utilizzata per stimare la funzione di densità di probabilità dei valori di intensità dei pixel nel tempo (PDF). Attraverso KDE è possibile discriminare i valori appartenenti al background da quelli associati al foreground, risultando efficace in contesti caratterizzati da variazioni complesse.
- **Metodi basati su Deep Learning**, i quali ricorrono a reti neurali profonde e modelli di apprendimento automatico per poter apprendere rappresentazioni complesse del background a partire da dataset di addestramento. Tali approcci risultano più robusti rispetto alle tecniche tradizionali, soprattutto in presenza di variazioni non lineari e scene altamente dinamiche.

Background in Microscopia

Nel contesto della microscopia, il background è generalmente costituito da segnali indesiderati che non sono direttamente correlati all'oggetto in esame, ma che possono interferire con la qualità e l'interpretazione dell'immagine. Tali segnali possono derivare da molteplici fonti, tra cui fenomeni ottici (ad esempio luce diffusa), rumore elettronico o delle superfici ottiche, caratteristiche intrinseche del campione biologico (come l'auto fluorescenza) e persino artefatti del sistema di imaging. Ovviamente anche in microscopia sussistono le stesse problematiche collegate al background nelle altre classi di immagini, ad esempio in caso di cellule di interesse (foreground) in movimento su scaffold o strutture considerate di sfondo.

Tipologie di background in Microscopia

È possibile classificare il background in microscopia in diverse categorie principali:

I. **Background Ottico**:
   a. *Luce diffusa e riflessa*: può derivare da superfici ottiche imperfette o da un'illuminazione non uniforme;
   b. *Aberrazioni ottiche*: difetti delle lenti o disallineamento del sistema di imaging che producono sfocature o immagini spurie;
   c. *Auto fluorescenza del substrato*: in microscopia a fluorescenza, alcuni materiali possono emettere fluorescenza intrinseca, sovrapponendosi al segnale utile.

II. **Background Digitale e Elettronico**:

a. *Rumore termico del sensore*: sensori CCD e CMOS generano un rumore intrinseco che può compromettere la qualità dell'immagine;

b. *Artefatti di acquisizione*: possono essere generati da fluttuazioni della tensione di alimentazione o da errori nel processo di digitalizzazione.

III. **Background Biologico**:

a. *Fluorescenza aspecifica*: in microscopia a fluorescenza, i coloranti possono legarsi in modo non specifico a componenti cellulari non target, creando segnali spuri;

b. *Movimenti del campione*: in varie tecniche di acquisizione immagini, come ad esempio la microscopia a contrasto di fase o DIC, fluttuazioni del mezzo o del campione possono introdurre variazioni indesiderate nell'immagine.

Tenuto conto di quanto appena riportato, per fornire un'idea più chiara, è riportata un'immagine associata alla ricostruzione di un tipico background dinamico in microscopia con corpuscoli e parti di cellule morte. Nell'immagine non vengono visualizzati microrganismi in movimento considerabili oggetti di foreground. Di conseguenza è possibile individuare solamente elementi di non rilevanza che costituiscono lo sfondo della scena.

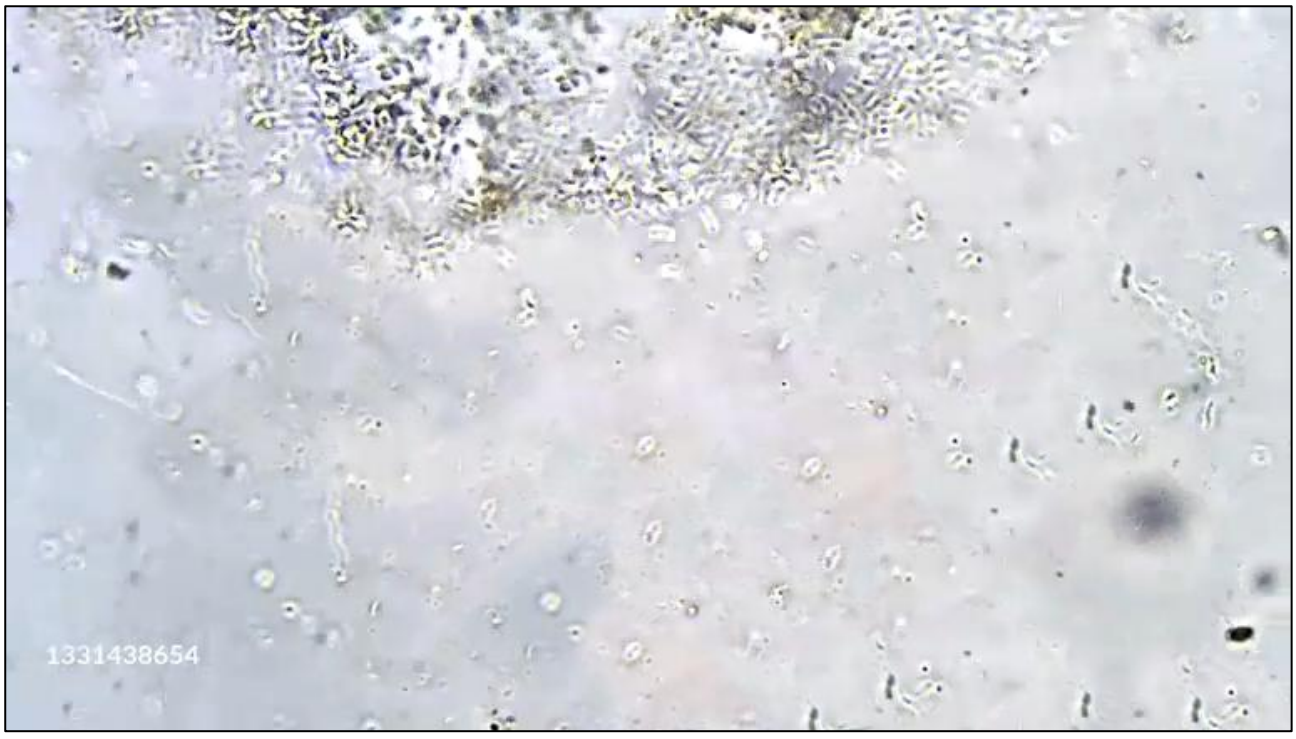

Figura 7: esempio di ricostruzione del background dinamico in un filmato di microscopia.

4 Parametri Video

4.1 Camera Statica

La camera statica rappresenta una modalità di acquisizione di video in cui la telecamera rimane fissa in una posizione invariata per l'intera durata della ripresa. Questo approccio è ampiamente utilizzato in diversi ambiti, quali il cinema, la videosorveglianza, la microscopia, l'animazione e le applicazioni di analisi automatica delle immagini. L'assenza di movimenti della telecamera semplifica l'interpretazione della scena e riduce la complessità computazionale delle operazioni di elaborazione video, in particolare nei processi di segmentazione e tracciamento del foreground.

Dal punto di vista percettivo e narrativo, l'uso di una camera statica influisce sulla composizione dell'inquadratura e sull'esperienza visiva dell'osservatore, consentendo di focalizzare l'attenzione sui soggetti in movimento o sugli eventi che si sviluppano all'interno di un contesto spaziale stabile.

Aspetti Tecnici della Camera Statica

- **Stabilità della Telecamera**

    - Per garantire una ripresa realmente statica, la telecamera deve essere posizionata su supporti adeguati, come treppiedi o superfici rigide, al fine di evitare vibrazioni o micro-movimenti.
    - In acquisizioni di lunga durata vengono spesso utilizzati supporti professionali, come teste fluide o sistemi di bloccaggio meccanico, che assicurano un'elevata stabilità nel tempo. Anche minime oscillazioni possono infatti compromettere l'analisi automatizzata, introducendo falsi movimenti nello sfondo.

- **Composizione dell'Inquadratura**

    - La composizione della scena ha un ruolo fondamentale anche in presenza di camera statica. Tecniche come la 'regola dei terzi' [18] permettono di bilanciare visivamente l'inquadratura, posizionando il soggetto principale in punti strategici;
    - La scelta della profondità di campo influisce sulla separazione visiva tra foreground e background; in particolare un'apertura ampia dell'obiettivo consente di mettere a fuoco il

soggetto lasciando lo sfondo sfocato, mentre un'apertura ridotta mantiene nitidi entrambi i piani;

- Anche la prospettiva e l'angolazione della telecamera influenzano la percezione della scena, trasmettendo sensazioni di neutralità, dominanza o vulnerabilità del soggetto ripreso.
    - Prospettiva a livello degli occhi: la visione risulta più neutra e naturale;
    - Angolazione dall'alto: il soggetto appare più debole o vulnerabile;
    - Angolazione dal basso: il soggetto appare dominante o potente.

- **Illuminazione**

    - L'illuminazione deve essere mantenuta il più possibile costante nel tempo. In presenza di luce naturale, è necessario considerare variazioni dovute a condizioni ambientali (ad esempio cambiamenti meteorologici o cicli giorno-notte);
    - Nel caso di illuminazione artificiale, è preferibile utilizzare sorgenti stabili e uniformi, evitando la formazione di ombre o riflessi che, con una camera statica, risultano particolarmente evidenti e possono interferire con i processi di analisi del foreground.

4.2 Background Statico

Come riportato in precedenza, il background statico è caratterizzato da uno sfondo che rimane invariato nel tempo per tutta la durata dell'acquisizione video. Questa condizione si verifica tipicamente in ambienti controllati, come riprese in interni, sistemi di videosorveglianza con telecamera fissa, esperimenti di laboratorio con acquisizioni in microscopia. La stabilità che si riscontra consente una più agevole separazione tra foreground e background, riducendo la complessità degli algoritmi di analisi e migliorando l'affidabilità dei risultati.

A livello computazionale, la presenza di uno sfondo statico rappresenta uno scenario favorevole per molte tecniche di segmentazione, da quelle meno recenti a quelle basate su deep learning, poiché le variazioni temporali osservate nella sequenza di immagini sono generalmente attribuibili solo agli elementi in primo piano.

Punti di attenzione nella gestione del background statico

Nonostante la relativa semplicità dello scenario, anche in presenza di background statico è necessario considerare alcuni aspetti critici:

- **Variazioni di illuminazione**: cambiamenti graduali o improvvisi della luce possono alterare i valori dei pixel dello sfondo, generando falsi positivi nella rilevazione del foreground;

- **Rumore del sensore**: il rumore elettronico può introdurre variazioni minime ma persistenti nello sfondo, influenzando la qualità della sottrazione;

- **Ombre e riflessi**: ombre proiettate dal foreground oppure riflessi su superfici lucide possono essere erroneamente classificati come elementi in primo piano;

- **Stabilità della telecamera**: anche lievi vibrazioni o micro-movimenti possono compromettere l'ipotesi di staticità dello sfondo, rendendo necessaria una fase di stabilizzazione preventiva.

Tecniche di gestione del background statico

In presenza di uno sfondo statico, è possibile adottare una serie di strategie inerenti alla sua gestione e modellazione:

1) **Sottrazione diretta dello sfondo**: tramite questa tecnica, lo sfondo viene stimato a partire da uno o più fotogrammi di riferimento, per poi essere sottratto dai frame successivi (solitamente le differenze significative nei valori dei pixel vengono associate al foreground). La procedura risulta semplice ed efficace in ambienti stabili, ma allo stesso tempo sensibile al rumore e a lievi variazioni di illuminazione.

2) **Modelli basati su media o mediana temporale**: in questa casistica il background viene stimato calcolando la media o la mediana dei valori di ciascun pixel su una finestra temporale. L'utilizzo della mediana consente di ridurre l'influenza degli oggetti in movimento, risultando particolarmente utile quando il foreground occupa solo una parte limitata della scena.

3) **Filtri spaziali e morfologici**: ricorrere all'utilizzo di filtri spaziali (ad esempio smoothing o filtri mediani) e di operazioni morfologiche, consente di ridurre il rumore residuo e di migliorare la procedura di separazione tra foreground e background, specialmente in presenza di piccoli artefatti.

4) **Approcci basati su soglia**: in scenari molto controllati, è possibile utilizzare soglie fisse o adattive sui valori di intensità per riuscire a distinguere con maggiore precisione il foreground dallo sfondo. Questa tipologia di approccio risulta particolarmente efficiente nel contesto della segmentazione video, ma al contempo è fortemente dipendente dalle condizioni di acquisizione.

4.3 Background Dinamico

Il background dinamico come anticipato nel capitolo precedente, è caratterizzato da variazioni temporali e spaziali dello sfondo dovute a diversi fattori, tra cui cambiamenti di illuminazione, presenza di elementi in movimento o movimenti della telecamera. In tali condizioni, i processi di segmentazione e di estrazione del foreground risultano significativamente più complessi, poiché le variazioni dello sfondo possono essere erroneamente interpretate come parte del primo piano.

Tra le principali cause che determinano un background dinamico si possono distinguere:

1. **Movimento della telecamera**: lo spostamento o la vibrazione della telecamera comportano una modifica continua dello sfondo, rendendo necessarie tecniche di stabilizzazione e allineamento dei fotogrammi per evitare errori nel rilevamento del foreground.

2. **Oggetti in movimento nello sfondo**: la presenza di persone, veicoli, animali o elementi naturali (come vento, onde, nuvole) introduce variazioni che complicano la separazione tra primo piano e sfondo.

3. **Variazioni di illuminazione:** cambiamenti di luce naturale o artificiale (ad esempio l'alba, il tramonto, oppure luci che si accendono e spengono) possono alterare in modo significativo l'aspetto dello sfondo, influenzando i valori di intensità dei pixel.

4. **Sfocatura e profondità di campo**: con una ridotta profondità di campo, lo sfondo può risultare sfocato, la cui apparenza varia in funzione della distanza degli oggetti dalla telecamera, introducendo ulteriori difficoltà nella modellazione dello sfondo.

Tecniche di gestione del background dinamico

I. **Rilevamento e sottrazione dello sfondo**: come già menzionato in precedenza, questa metodologia prevede l'utilizzo di un fotogramma iniziale, il quale viene confrontato con i frame successivi; se un pixel cambia significativamente rispetto al frame di riferimento, allora viene considerato parte del foreground. Il limite di questo approccio ricade nell'identificazione del primo piano nel momento in cui si manifestano piccole variazioni dello sfondo; in tale situazione non avviene una corretta distinzione tra foreground e background e, l'analisi risulta fallimentare. A fronte di ciò sono stati sviluppati modelli statistici come il *GMM* per migliorare la precisione, calcolando una media di valori di colore e luminosità in modo da comprendere gli elementi che appartengono correttamente allo sfondo.

II. **Stabilizzazione dello sfondo**: in presenza di movimenti della telecamera, sono impiegati algoritmi di estrazione di feature come lo *Scale-Invariant-Feature Transform* (SIFT) [19] o lo *Speeded – Up Robust Features* (SURF) [20] che mirano ad individuare punti stabili nella scena per poi utilizzarli come riferimento per la compensazione del movimento.

III. **Filtri e tecniche di separazione del foreground**: ulteriori strategie prevedono di distinguere e risaltare punti specifici della scena, queste includono la *Stima della profondità*, *Filtri Motion blur* o tecniche basate su movimento per enfatizzare il foreground e ridurre l'impatto delle variazioni dello sfondo.

A titolo di esempio, nel seguito viene mostrato un fotogramma tratto da un'acquisizione video outdoor, in cui lo sfondo assume un comportamento dinamico dovuto al movimento delle foglie presenti nella scena e retrostanti ai soggetti principali in primo piano. Questo esempio evidenzia le difficoltà tipiche associate alla gestione di un background dinamico e la necessità di tecniche di analisi più robuste.

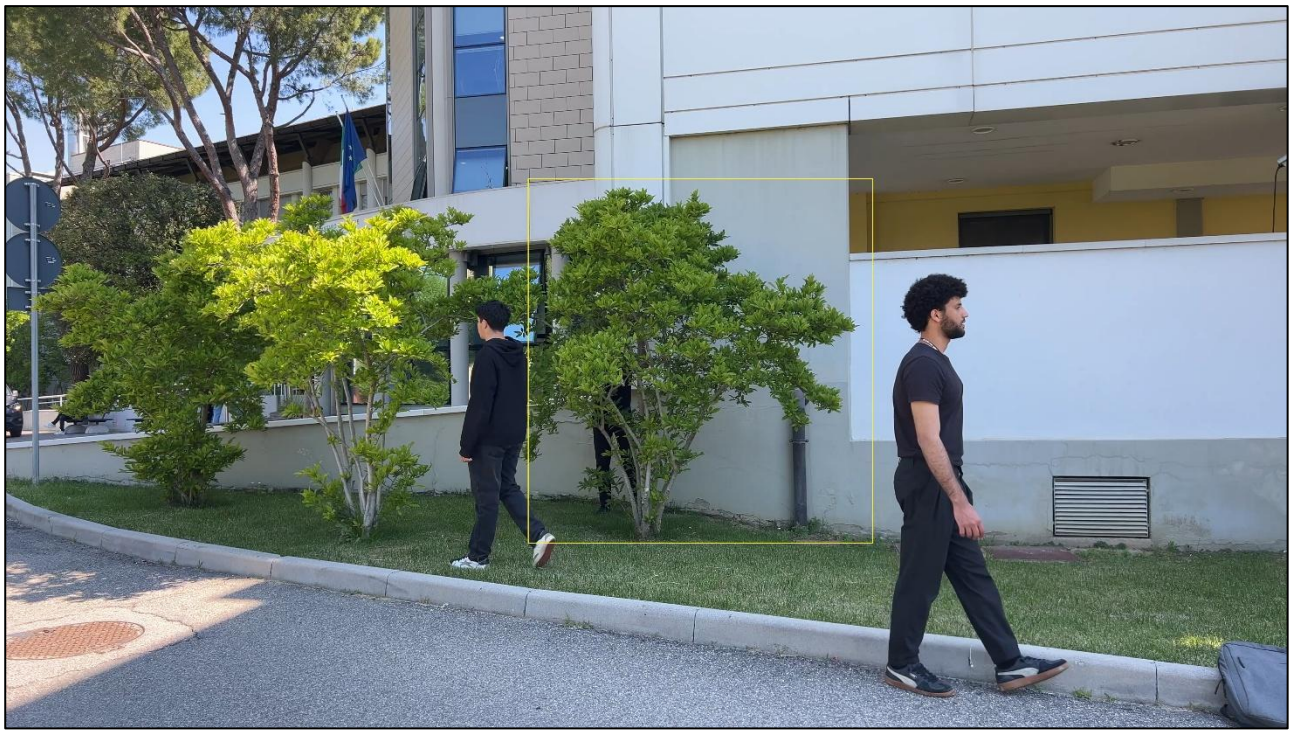

Figura 8: esempio di un fotogramma estratto a circa metà del video originale con acquisizione outdoor.

5 Materiali e Metodi

La presente sezione descrive i materiali utilizzati per la realizzazione del lavoro sperimentale, con particolare riferimento all'attrezzatura impiegata per l'acquisizione video, alle risorse hardware servite per l'elaborazione dei dati e al dataset di sequenze video analizzate. La scelta dei materiali e dell'attrezzatura è stata effettuata con l'obiettivo di garantire una raccolta dati eterogenea, rappresentativa di differenti condizioni di acquisizione e scenari applicativi, al fine di valutare in modo sistematico le prestazioni delle tecniche di segmentazione semiautomatica adottate.

5.1 Attrezzatura per l'acquisizione video

Per la registrazione dei filmati di videosorveglianza è stato utilizzato il modello di smartphone *iPhone 13 Pro*, scelto per la qualità del sensore di acquisizione, la stabilità dell'immagine e la possibilità di registrare video in alta risoluzione con un frame rate costante. Di seguito sono riportate le principali specifiche tecniche riguardo al dispositivo adoperato:

- **Chip**:
    - A15 Bionic;
    - CPU 6-core con 2 performance core e 4 efficiency core;
    - Neural Engine 16-core.
- **Fotocamera**:
    - Sistema di fotocamere Pro da 12MP con teleobiettivo ($f/2.8$), grandangolo e ultra-grandangolo;
    - Zoom in ottico 3 × con estensione totale 6 × e zoom digitale fino a 15 ×;
    - Formati immagini acquisiti: HEIF e JPEG.
- **Registrazione video**:
    - Registrazione video a 4K a 30 fps, anche in versione macro;
    - Stabilizzazione ottica dell'immagine video tramite sensore;
    - Formato dei filmati registrati: MOV.
- **Sistema operativo**: IOS 15.

Lo stesso modello di smartphone utilizzato per i filmati di videosorveglianza è stato impiegato anche per la registrazione dei video inerenti all'analisi macroscopica, relativi all'osservazione dello spostamento di insetti.

Al fine di garantire condizioni di ripresa controllate e ridurre al minimo eventuali vibrazioni o movimenti indesiderati nei filmati con camera statica, lo smartphone è stato montato su un treppiede, utilizzato come supporto fisso durante l'intera durata delle acquisizioni.

5.2 Luoghi e sorgenti dei dati video

Le sequenze video utilizzate nel presente lavoro provengono sia da acquisizioni dirette sia da sorgenti esterne, selezionate in modo da valutare il comportamento del software di segmentazione semiautomatica in differenti scenari di interesse.

- **Filmati di videosorveglianza**: i video sono stati acquisiti presso l'*Istituto Romagnolo per lo Studio dei Tumori "Dino Amadori"* (IRST) di Meldola (FC) in differenti condizioni ambientali, al fine di includere sia casi di background statico sia di background dinamico. In particolare, i video caratterizzati da sfondo statico sono stati registrati all'interno dell'aula 'Tison' dell'Istituto, un ambiente indoor controllato, privo di variazioni significative di illuminazione e di elementi di disturbo in movimento. Al contrario, i filmati con sfondo dinamico sono stati acquisiti all'esterno della struttura, in un'area caratterizzata dalla presenza di alberi e vegetazione. In questo scenario outdoor, lo sfondo è soggetto a variazioni temporali dovute al movimento delle foglie, cambiamenti di illuminazione naturale e possibili ombre dinamiche.

- **Filmati per analisi macroscopica**: le sequenze sono state acquisite in ambiente controllato, al fine di osservare il movimento di uno o più soggetti in condizioni di parziale o totale esposizione, con dinamiche continue o intermittenti. Per questa categoria di filmati sono stati registrati video con background statico, mantenendo lo smartphone stabilizzato.

- **Filmati di microscopia**: per l'analisi delle sequenze microscopiche è stato utilizzato un singolo video reperito dalla piattaforma *Getty Images* (disponibile al seguente link: https://www.gettyimages.it/detail/video/micro-organisms-floating-in-dirty-water-filmati-stock/1331438654, con licenza *Royalty-Free)*, raffigurante microorganismi in sospensione in

un mezzo liquido. Tale sorgente è stata scelta per la qualità delle immagini e per la presenza sia di foreground che background dinamici.

Considerato quanto sopra esposto, è quindi riportato un insieme di tre immagini corrispondenti ai fotogrammi di ciascuna categoria di filmato presa in considerazione.

Figura 9: esempio di un fotogramma per la classe dei filmati di videosorveglianza.

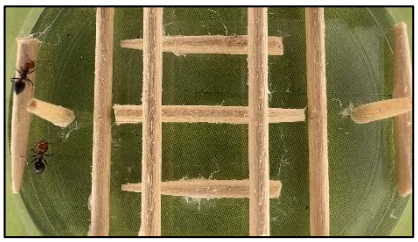

Figura 10: esempio di un fotogramma per la classe dei filmati di analisi macroscopica.

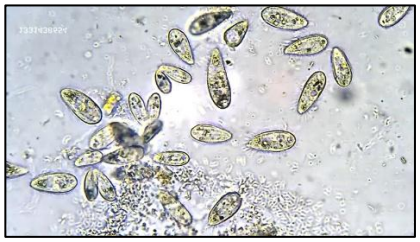

Figura 11: esempio di un fotogramma associato alla classe dei filmati di microscopia.

5.3 Attrezzature hardware per l'elaborazione dei dati

A proposito dell'elaborazione dei video e del testing del software di segmentazione semiautomatica, questi ultimi sono stati eseguiti utilizzando un computer portatile modello *Dell Latitude 3450*. Tale sistema è stato impiegato per tutte le fasi del pre-processing, segmentazione e analisi dei risultati, garantendo uniformità nelle condizioni di esecuzione degli algoritmi. I dettagli che seguono riportano la scheda tecnica del computer utilizzato:

- **Processore**: Intel(R) Core(TM) Ultra 5 125U (3.60 GHz);
- **Memoria RAM installata**: 16,0 GB;
- **Tipo di sistema operativo**: Sistema operativo a 64 bit, processore basato su x64
    - edizione: Windows 11 Pro;
    - versione: 25H2;
- **Scheda Video**: NVIDIA GeForce Nvidia GeForce 830M (DDR3 da 2 GB);
- **GPU**: Intel(R) Graphics;
- **NPU**: Intel(R) AI Boost.

5.4 Tipologia di soggetti filmati durante le acquisizioni video

Per quanto riguarda i soggetti ripresi durante le registrazioni, essi variano in funzione della tipologia di filmato e dello scenario di acquisizione considerato. In particolare, la natura del soggetto, il suo comportamento dinamico e il grado di interazione con l'ambiente circostante dipendono strettamente dal contesto applicativo del video, influenzando in modo significativo le caratteristiche del foreground e, di conseguenza, il processo di segmentazione.

- Nei filmati di videosorveglianza, i soggetti ripresi in movimento all'interno di tali sequenze sono individui umani normodotati, impiegati come elementi dinamici della scena esclusivamente a fini sperimentali e di analisi.

- Nel contesto specifico dell'analisi macroscopica, gli esemplari utilizzati per le registrazioni sono formiche (*Formicidae*), raccolte manualmente in un ambiente naturale non controllato, corrispondente ad un giardino privato, senza l'utilizzo di esche o sostanze chimiche. La raccolta è avvenuta in condizioni ambientali naturali e, gli insetti non sono stati sottoposti ad alcun trattamento chimico o meccanico prima delle riprese e sono stati mantenuti per un periodo limitato esclusivamente ai fini della registrazione. Per questa tipologia di filmati è stata impiegata una *Piastra di Petri*, utilizzata come ambiente controllato per delimitare l'area di movimento dei soggetti così da facilitare l'analisi visiva.

- Nell'ambito della microscopia, i "soggetti" osservati corrispondono a *Parameci*, ovvero un genere di protisti infusori protozoi appartenenti alla classe dei ciliati. Tali microrganismi si trovano in movimento all'interno di una soluzione acquosa torbida, con presenza di particelle, residui organici e altri elementi. Essi costituiscono la parte di maggiore rilevanza all'interno della scena e, sono ben distinti dallo sfondo grazie ad un contorno particolarmente definito.

5.5 Dataset dei video acquisiti

Il dataset utilizzato nel lavoro è composto da un insieme di sequenze video differenziate in base a diversi parametri, quali la staticità o dinamicità della telecamera, il numero di oggetti presenti nella scena e le modalità di movimento ed esposizione del foreground.

Al fine di classificare in modo sistematico le sequenze di filmati di videosorveglianza, è stata adottata la seguente legenda:

- S/D: per indicare rispettivamente la camera statica o dinamica;
- O (Object): per indicare il soggetto di riferimento;
- 1/2/3: numero di soggetti in movimento presenti nella scena;

Classificazione dei casi di studio:

1. Movimento continuo con soggetto totalmente esposto;
2. Movimento interrotto con soggetto totalmente esposto;
3. Movimento continuo con soggetto parzialmente esposto;
4. Movimento interrotto con soggetto parzialmente esposto;
5. Un soggetto con movimento continuo e totalmente esposto; due soggetti con movimento continuo e parzialmente esposti;
6. Un soggetto con movimento continuo e totalmente esposto; due soggetti con movimento interrotto e parzialmente esposti;
7. Un soggetto con movimento interrotto e totalmente esposto; due soggetti con movimento continuo e parzialmente esposti;
8. Un soggetto con movimento interrotto e totalmente esposto; due soggetti con movimento interrotto e parzialmente esposti;

Un ulteriore legenda, simile alla precedente è stata adoperata per la classificazione dei video contenenti gli insetti in movimento:

- S/D: per indicare rispettivamente la camera statica o dinamica;
- A (Ant): per indicare il soggetto di riferimento (corrispondente alla formica in esame);
- 1/2/3: numero di soggetti in movimento presenti nella scena;

Classificazione dei casi di studio:

1. Uno o più soggetti totalmente esposti in movimento continuo;
2. Uno o più soggetti parzialmente visibili in movimento continuo;

In riferimento al campo della microscopia, come già menzionato in precedenza, è stato considerato solamente un video rappresentante una serie di microrganismi in movimento, con distribuzione non uniforme. Date le proprietà variabili dovute a distribuzioni non uniformi, spostamenti traslazionali e rotazionali, velocità variabili e perdite di visualizzazione dei soggetti, il filmato è stato etichettato con il titolo *Microorganism_Motility_01*.

Le classificazioni riportate rendono la struttura dati ordinata, permettendo di analizzare in modo strutturato il comportamento dell'algoritmo di segmentazione in presenza di condizioni progressivamente più complesse, valutandone la robustezza rispetto a variazioni nel movimento, nell'esposizione del foreground e nel numero di soggetti presenti nella scena.

5.6 Dettagli tecnici sui video acquisiti

Sulla base delle legende adottate, sono stati considerati i video classificati come *S1O1* e *S2O2* nel contesto della videosorveglianza, ed il video intitolato *S2A2* nello scenario delle valutazioni macroscopiche. Per il contesto della microscopia è sempre stato considerato il filmato *Microorganism_Motility_01*.

I filmati di videosorveglianza presentano le seguenti specifiche tecniche:

I. Filmato *S1O1*:
- Risoluzione: 3840 × 2160 px;
- Frame Rate: 30 fps;
- Durata: 4,65 secondi;
- Numero totale di frame: 140.

II. Filmato *S2O2*:
- Risoluzione: 3840 × 2160 px;
- Frame Rate: 30 fps;
- Durata: 6,75 secondi;
- Numero totale di frame: 203.

Il filmato di analisi macroscopica *S2A2* è caratterizzato dai seguenti parametri:

- Risoluzione: 2160 × 3840 px ;
- Frame Rate: 30 fps;
- Durata: 5,63 secondi;
- Numero totale di frame: 169.

Il video inerente al campo della microscopia *Microorganism_Motility_01* possiede i seguenti dettagli:

- Risoluzione: 768 × 432 px;
- Frame Rate: 30 fps;
- Durata: 4,80 secondi;
- Numero totale di frame: 144.

5.6 Descrizione dei dataset sperimentali

Come già anticipato, le quattro tipologie di filmati selezionati, coprono scenari variabili, sia in termini di tipologia di acquisizione sia per quanto riguarda le caratteristiche dei soggetti in movimento, la complessità dello sfondo e la luminosità dell'ambiente. La scelta dei video è guidata dall'obiettivo di testare la robustezza e la versatilità dell'approccio di segmentazione, evidenziando i punti di forza e limiti dell'algoritmo in situazioni realistiche. Per ciascun filmato sono approfondite le principali caratteristiche e le motivazioni che ne hanno determinato l'inclusione nel protocollo sperimentale.

1. **Filmati di videosorveglianza**: questi presentano soggetti umani normodotati in movimento all'interno della scena. Tali video sono stati acquisiti con camera stabilizzata e mostrano una dinamica tipica dei sistemi di monitoraggio, il cui movimento dei soggetti costituisce l'elemento informativo principale, mentre lo sfondo risulta prevalentemente statico. La scelta di questi filmati è motivata dalla possibilità di verificare l'efficacia dell'algoritmo nella corretta discriminazione del primo piano in presenza di ombre, parziali occlusioni o interruzioni nel movimento. Inoltre, sono stati considerati i casi di studio specifici *S1O1* e *S2O2* già affrontati in letteratura [21] o in contesti sperimentali analoghi.

    a. Filmato *S1O1*: è caratterizzato da uno sfondo statico ed una scena relativamente semplice con un solo soggetto in movimento continuo. Le condizioni di illuminazione risultano uniformi e non sono presenti occlusioni che coinvolgono il foreground. La scelta di questo filmato è motivata dalla volontà di disporre un caso di riferimento

controllato, utile per valutare l'algoritmo in condizioni favorevoli, con separazione tra primo piano e sfondo ben definita.

b. Filmato *S2O2*: presenta una scena più complessa, caratterizzata dalla presenza di due soggetti in movimento, con una notevole occlusione fra di essi. Questo elemento introduce un'importante ambiguità nella distinzione degli elementi, rendendo la segmentazione automatica più critica. Tale filmato è quindi stato scelto a causa della necessità di valutare la robustezza dell'algoritmo in condizioni operative reali, con una rilevante sovrapposizione dei soggetti.

2. **Filmati di analisi macroscopica**: in particolare il video *S2A2* inquadra due formiche in movimento all'interno della *Piastra di Petri*, una delle quali è caratterizzata da diverse occlusioni. La selezione di tale filmato è stata guidata dalla necessità di testare il modello in uno scenario biologico reale, particolarmente complicato. Quest'ultimo infatti, è costituito da soggetti di piccole dimensioni, movimenti rapidi, spostamenti irregolari e frequenti interazioni con gli elementi predisposti in diverse zone, corrispondenti a bacchette di legno posizionate all'interno della 'piastra'. L'insieme di questi aspetti rende la segmentazione particolarmente critica, ma permette di guidare tale processo ricorrendo alle diverse modalità offerte dal modello *SAM*.

3. **Filmati di microscopia**: il video *Microorganism_Motility_01* rappresenta un caso di microscopia centrale, in cui i soggetti ripresi sono caratterizzati da spostamenti rapidi, per cui la loro visualizzazione risulta altamente variabile all'interno della ripresa. Questo filmato introduce quindi notevoli complessità che non sono presenti nelle altre categorie di video acquisiti. La scelta di includere questo filmato risponde all'esigenza di analizzare il comportamento dell'algoritmo in ambito biomedicale, dove la segmentazione di entità con dimensioni molto ridotte (quali microrganismi, tessuti, virus, batteri, etc…) è un requisito fondamentale.

6 Algoritmo Foreground Detection

Per la realizzazione dell'algoritmo di *Foreground Detection* è stato impiegato l'ambiente di programmazione *MATLAB*, scelto per la sua efficacia nella gestione di operazioni matematiche complesse, nell'elaborazione dei segnali e immagini digitali e nella visualizzazione dei risultati tramite grafici e rappresentazioni dedicate. In particolare, *MATLAB* consente di operare in modo naturale su strutture matriciali e tensori, caratteristica che risulta particolarmente adatta alla rappresentazione e manipolazione dei fotogrammi video, trattati come matrici di intensità o come array multidimensionali nel caso di immagini a colori.

Allo stesso tempo, l'interfaccia di programmazione è stata utilizzata come piattaforma di supporto per l'integrazione di tecniche di segmentazione semiautomatica del foreground come *SAM*. Tale integrazione permette di combinare approcci tradizionali di elaborazione delle immagini con metodologie avanzate, migliorando la precisione e la solidità della segmentazione del primo piano in scenari caratterizzati da un elevata variabilità del contenuto visivo.

6.1 SAM

Come anticipato nei capitoli precedenti, il *Segment Anything Model* si configura come *foundation model*, ovvero come modello di intelligenza artificiale pre-addestrato su una vasta quantità di dati eterogenei, progettato per essere adattabile a molteplici compiti di segmentazione senza la necessità di un riaddestramento completo. L'approccio adottato è di tipo generativo, in grado di produrre maschere di segmentazione a partire da input variabili forniti dall'utente.

La valutazione delle prestazioni di questi modelli richiede l'impiego di metriche precise, in grado di misurare in modo accurato la qualità delle segmentazioni ottenute. Tra le più utilizzate vi sono il *Dice Similarity Coefficient* (DSC) [22], la *Hausdorff Distance* [23] e il *Jaccard Similarity Index* [24], definito come il rapporto tra l'area di intersezione e l'area di unione tra la segmentazione stimata e quella di riferimento.

Dal punto di vista dell'architettura, *SAM* si basa sui *Vision Transformer* (ViT) [25], in particolare sulle varianti *ViT-B* (Base) e *Vit-H* (Huge), che differiscono per numero di parametri, profondità della rete e capacità di rappresentazione. Questo tipo di architettura propone di suddividere l'immagine in porzioni regolari, per poi trattarle come elementi discreti di una sequenza. Mediante questa

impostazione, il modello è in grado di cogliere il contesto globale e le dipendenze a lungo raggio tra diverse regioni dell'immagine fin dalle prime fasi di elaborazione, evitando la necessità di costruire tali relazioni attraverso numerosi livelli convoluzionali, come avviene nelle *Convolutional Neural Network* (CNN) [26].

Architettura di SAM

Alla luce di quanto esposto, fornita in ingresso un'immagine, il modello *SAM* è in grado di individuare e isolare automaticamente uno o più elementi d'interesse, quali oggetti, persone o strutture specifiche, producendo una maschera che identifica con precisione i pixel appartenenti al foreground.

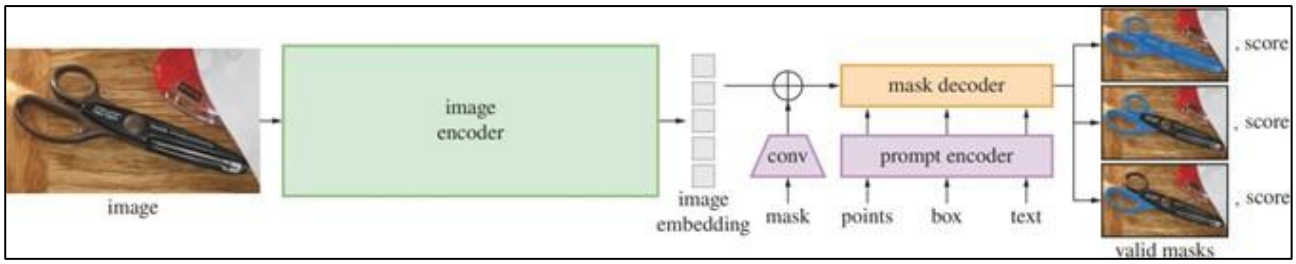

Figura 12: Rappresentazione schematizzata dell'architettura SAM.

In termini strutturali, *SAM* è composto da tre moduli principali, ciascuno con un ruolo ben definito nel processo di segmentazione: Image Encoder, Prompt Encoder e Mask Decoder. La separazione funzionale di questi componenti, consente al modello di combinare una rappresentazione globale dell'immagine con indicazioni fornite dall'utente.

Image Encoder

Questo primo componente ha il compito di analizzare l'intera immagine di input in modo da generare una rappresentazione numerica compatta, comunemente definita *embedding* [27]. Tale rappresentazione cattura le caratteristiche visive più rilevanti dell'immagine, come forme, contorni, texture e relazioni spaziali tra le diverse regioni. L'Image Encoder è implementato mediante un *ViT*, per cui l'immagine è suddivisa in un insieme di *patch* di dimensioni fisse, ciascuna delle quali è convertita in un vettore numerico. Tramite un meccanismo di 'auto-attenzione', il *ViT* è in grado di mettere in relazione tutte le patch tra loro, apprendendo una rappresentazione astratta e informativa dell'intera scena. Il modulo Encoder ha la capacità di operare su immagini ad alta risoluzione,

fornendo come output un array multidimensionale (tensore), che funge da base informativa per le fasi successive del processo di segmentazione.

Prompt Encoder

Il secondo membro dell'architettura, ha l'obiettivo di tradurre le indicazioni fornite dall'utente, denominate *prompt*, in una rappresentazione numerica comprensibile al modello. I prompt consentono di guidare la segmentazione e possono assumere diverse forme, come punti, riquadri o meglio definiti *bounding box* [28] e descrizioni testuali.

- Per quanto riguarda i prompt spaziali (punti o riquadri), le coordinate vengono arricchite con informazioni di codifica posizionale, in modo da preservare la relazione spaziale con l'immagine. Nello specifico, le bounding box isolano oggetti specifici all'interno di un fotogramma video, consentendo ai sistemi di intelligenza artificiale di conoscere con precisione il posizionamento di un elemento, piuttosto che limitarsi alla sola esistenza nella scena.

- Nel caso di prompt testuali, viene utilizzato un modello pre-addestrato per l'allineamento tra linguaggio e immagini, come *CLIP* (Contrastive Language Image Pretraining) [29], che consente di convertire il testo in una rappresentazione numerica semanticamente coerente con il contenuto visivo. Questo tipo di integrazione consente a *SAM* di supportare prompt in diverse modalità, aumentando la flessibilità del processo.

Mask Decoder

Quest'ultimo modulo costituisce la fase finale del modello, in grado di utilizzare congiuntamente l'embedding dell'immagine prodotto dall'Image Ecoder, e i prompt codificati dal Prompt Encoder per generare maschere di segmentazione. Il decoder sfrutta un 'meccanismo di attenzione' per focalizzarsi sulle regioni dell'immagine più rilevanti rispetto al prompt fornito, guidando il modello alla selezione delle aree che corrispondono all'oggetto di interesse. Il processo di generazione della maschera avviene in modo interattivo, raffinando gradualmente i contorni e migliorando la precisione della segmentazione ad ogni passaggio.

L’output finale del Mask Decoder è una o più maschere binarie, in cui i pixel etichettati con valore numerico 1 appartengono all’oggetto segmentato, mentre quelli con valore 0 rappresentano lo sfondo. In alcuni casi, il modello può produrre più maschere alternative associate a un punteggio di confidenza, offrendo all’utente la possibilità di selezionare il risultato più appropriato.

Data Engine

La creazione di modelli di segmentazione accurati è fortemente limitata dalla disponibilità di maschere in alta qualità, che risultano più costose e complesse da ottenere rispetto ad altre forme di annotazione visiva. Per superare tale limite, è stato sviluppato un *data engine* corrispondente ad un sistema di raccolta dati scalabile che ha permesso la creazione del dataset *SA-1B*, composto da 1,1 miliardi di maschere di segmentazione.

Il motore di dati si articola in tre fasi principali, caratterizzate da un progressivo aumento del livello di automazione:

1. **Assisted-manual stage**: in questa prima fase le maschere sono state create manualmente da annotatori esperti mediante uno strumento interattivo basato su browser. Sono stati selezionati i pixel appartenenti agli oggetti di interesse, per poi perfezionare le maschere utilizzando strumenti di editing a livello di singolo pixel. Non sono stati imposti vincoli rigidi e sono stati annotati sia oggetti sia regioni amorfe, seguendo un ordine di importanza visiva. Durante questa fase, il modello è stato riaddestrato più volte, migliorando le sue prestazioni e riducendo il tempo medio di annotazione per maschera da 34 a 14 secondi, raccogliendo un totale di circa 4,3 milioni di maschere a partire da 120.000 immagini.

2. **Semi-automatic stage**: per questo secondo stadio l’obiettivo è stato aumentare la diversità delle maschere annotate. Un rilevatore di oggetti generico, addestrato sui dati della fase precedente, ha generato in modo automatico un insieme preliminare di maschere che sono state in seguito presentate agli annotatori per identificare gli oggetti mancanti. Questo approccio ha consentito di concentrare l’intervento umano sugli elementi più complessi o meno evidenti, aumentando il numero di maschere per immagine. Per questa fase sono state raccolte 5,9 milioni di maschere in 180.000 immagini, arrivando ad un totale di 10,2 milioni. Inoltre a causa della complessità degli oggetti rimanenti il tempo medio di annotazione è

tornato a 34 secondi, ma con un aumento del numero medio di maschere da 44 a 72 per immagine.

3. **Fully-automatic stage**: in quest'ultima fase, l'annotazione è diventata completamente automatica grazie all'elevata accuratezza raggiunta dal modello e ad una gestione esplicita delle ambiguità. In presenza di prompt incerti, *SAM* è in grado di generare maschere valide per un singolo input, ciascuna associata a un punteggio di confidenza che rappresenta una stima dell'*Intersection over Union* (IoU) [30] rispetto alla maschera di riferimento. Il modello genera proposte di segmentazione a partire da una griglia regolare sovrapposta all'immagine, per poi selezionare in seguito le maschere più stabili e accurate eliminando quelle ridondanti.

Dal punto di vista computazionale, l'architettura *SAM* è stata progettata con una forte attenzione all'efficienza. Una volta calcolato l'embedding dell'immagine, il prompt encoder e il mask decoder possono essere eseguiti in tempi molto ridotti, di circa 50 millisecondi su CPU, consentendo un'interazione fluida e in tempo reale.

Dataset SA-1B

Il risultato descritto nella sezione di 'Data Engine' equivale al dataset *SA-1B* (Segment Anything – 1 Billion), uno dei più grandi dataset di segmentazione attualmente disponibili. Esso è composto da circa 11 milioni di immagini ad alta risoluzione e da 1,1 miliardi maschere di segmentazione, tutte ottenute mediante il motore di elaborazione dati sviluppato per *SAM*. Le immagini sono state concesse in licenza da un fornitore che collabora direttamente con i fotografi, garantendo il rispetto delle normative sulla privacy. Le immagini originali presentano una risoluzione media elevata di circa 3300 × 4950 pixel, significativamente superiore a quella di molti dataset comunemente utilizzati nella visione artificiale. Per facilitare l'accessibilità e ridurre l'impatto in termini di archiviazione e gestione dati, è stata rilasciata anche una versione sottocampionata delle immagini, con lato corto fissato a 1550 pixel.

Il set di dati *SA-1B* è stato reso pubblico con l'obiettivo di favorire lo sviluppo di *foundation model* per la visione artificiale, offrendo una base dati ampia, diversificata e di alta qualità.

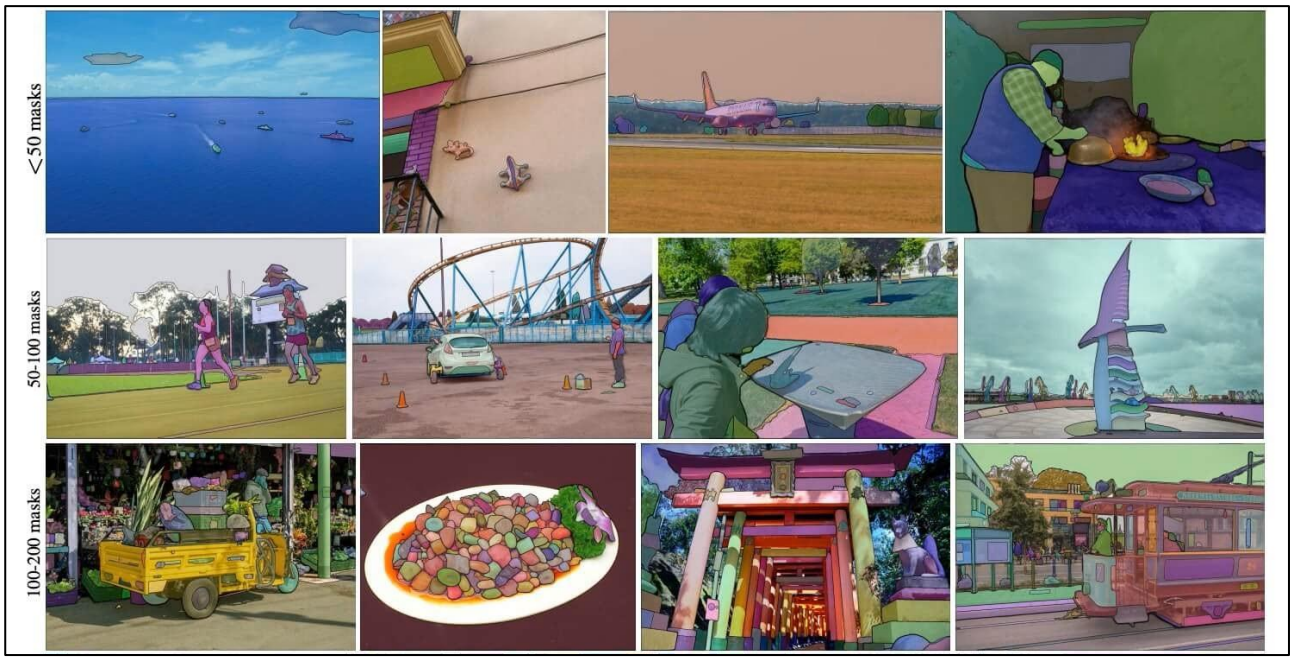

Figura 13: Esempi di immagini tratte dal dataset SA-1B con maschere sovrapposte (manuali e automatiche) distinte da un colore univoco.

SAM nell'imaging medicale

Il modello ha dimostrato elevate capacità nella segmentazione di immagini naturali riuscendo a produrre maschere accurate anche in assenza di un addestramento specifico. Tuttavia l'applicazione diretta di *SAM* alle immagini mediche presenta diverse criticità. La caratteristica di *prompt-based* del modello, rende il suo utilizzo non immediatamente sovrapponibile ai flussi di lavoro dell'imaging medicale, dove l'oggetto da segmentare non è facilmente distinguibile e possono essere presenti numerose strutture sovrapposte.

Nonostante ciò, in letteratura sono stati individuati diversi approcci per integrare *SAM* nel processo di segmentazione delle immagini mediche.

- Un primo ambito di applicazione riguarda l'annotazione semiautomatica con approccio *human-in-the-loop*, in cui l'utente fornisce prompt iniziali e il modello genera una maschera preliminare che può essere accettata o corretta iterativamente. In alternativa, *SAM* può operare in modalità *segment everything*, generando automaticamente un insieme di maschere sull'intera immagine, che vengono successivamente selezionate e modificate dall'operatore umano.

- Un secondo approccio è rappresentato dall'*input augmentation* [31], il quale consiste nell'arricchire le immagini mediche grezze con informazioni aggiuntive prima di sottoporle al modello di segmentazione. In particolare, le maschere generate da *SAM* sono impiegate come dettagli aggiuntivi in ingresso ai modelli di segmentazione specificamente progettati per l'imaging medicale.

Questo approcci consentono di arricchire i dati, guidare l'attenzione del modello verso le regioni di interesse e sfruttare le conoscenze pregresse di *SAM*, migliorando così le prestazioni complessive e la capacità di generalizzazione sui dati medici.

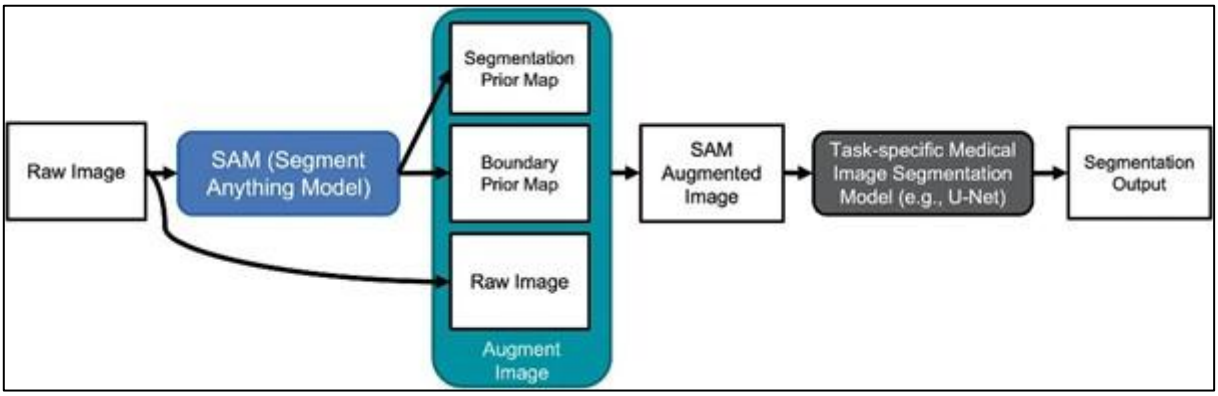

Figura 14: Rappresentazione schematizzata dell'approccio tramite Input Augmentation.

6.2 Descrizione dell'algoritmo

L'algoritmo sviluppato, in continuità con i concetti introdotti nelle sezioni precedenti, è concepito per operare su diverse tipologie di video di immagini acquisiti come sequenze di fotogrammi. In particolare, i video utilizzati come input sono caratterizzati da camera stabilizzata, in presenza di condizioni di background sia statico che dinamico.

L'obiettivo principale consiste nell'analizzare la sequenza video fotogramma per fotogramma al fine di individuare, segmentare e isolare automaticamente gli elementi appartenenti al foreground. Il risultato del processo è la generazione di un numero di maschere di segmentazione pari al numero complessivo di fotogrammi che compone il video originale, in modo da poterle applicare in seguito tramite l'interfaccia del software *ViFoSe*.

L'intero sistema è stato sviluppato utilizzando *MATLAB App Designer*, integrando funzionalità di interfaccia grafica, gestione dei dati, deep learning e image processing. In questa sezione viene fornita una descrizione dell'intero codice, illustrandone l'architettura, le scelte progettuali e il flusso operativo.

Inizializzazione e caricamento dei dati

Il processo ha inizio con la selezione da parte dell'utente delle cartelle contenenti rispettivamente i fotogrammi del video d'interesse (in formato PNG) e di destinazione per il salvataggio delle maschere di segmentazione; le immagini sono ordinate numericamente per assicurare la corretta sequenzialità temporale. I file sono organizzati in un array di strutture denominato *imageFiles*, che contiene tutti i file PNG presenti nella cartella selezionata. Il primo frame viene individuato tramite l'indice iniziale dell'array e il suo percorso assoluto viene costruito tramite la funzione *fullfile*, che combina il percorso della cartella e il nome del file. L'immagine così individuata viene quindi caricata in memoria tramite *imread*.

firstImagePath = fullfile(imageFiles(1).folder, imageFiles(1).name);

Prima dell'inizializzazione del modello, l'utente ha la possibilità di inserire manualmente il valore corrispondente ad una serie di parametri che sono impiegati nelle fasi successive di filtraggio e raffinamento morfologico. Tra questi si distinguono rispettivamente:

- La **Dimensione minima d'oggetto** (*MinimumObjectSize*), la quale definisce la soglia minima di area (in pixel) che una regione segmentata deve possedere per essere considerata un oggetto valido. Formalmente, dato un insieme di componenti connesse C_i ottenute dalla maschera binaria, una regione viene mantenuta se l'area della regione i è maggiore o uguale al valore impostato manualmente:

  $$Area(C_i) \geq A_{min}$$

  Questo parametro è particolarmente utile per la rimozione di oggetti troppo piccoli generati da errori di segmentazione, rumore, falsi positivi e artefatti locali dell'immagine.

- Il **Fattore di scala** (*ScaleFactor*), il quale coincide con un coefficiente moltiplicativo utilizzato per stimare automaticamente la dimensione minima degli oggetti a partire dalla segmentazione manuale iniziale. Dopo la selezione manuale degli oggetti, viene calcolata l'area media:

$$\overline{A} = \frac{1}{N}\sum_{i=1}^{N} Area(C_i)$$

Il valore effettivo *MinimumObjectSize* utilizzato dall'algoritmo diventa:

$$A_{min} = round(\overline{A} \times ScaleFactor)$$

Lo *ScaleFactor* consente di adattare automaticamente la soglia minima alla scala reale degli oggetti e rendere il metodo invariante alla risoluzione dell'immagine; nel codice viene applicato dopo la segmentazione manuale iniziale e dopo ogni fase di correzione manuale.

- L'**Area del fattore di scala** (*AreaScalingFactor*), la quale introduce una soglia dinamica dipendente dalla dimensione dell'immagine (con H corrispondente all'altezza e W alla larghezza), utilizzata per decidere se applicare o meno operazioni morfologiche. La soglia viene calcolata come:

$$A_{dyn} = AreaScalingFactor \times H \times W$$

Il valore medio delle aree segmentate viene confrontato con A_{dyn} e, nel caso in cui l'area media (*meanArea*) risulta maggiore di A_{dyn} allora sono applicate le operazioni morfologiche (*imclose*, *imfill*). Questo meccanismo introduce un comportamento adattivo che evita l'*over-processing* di oggetti piccoli e applica il raffinamento solo quando gli oggetti sono sufficientemente estesi.

- Il **Raggio del disco** (*DiskRadius*), che definisce il raggio in pixel dell'elemento strutturante circolare utilizzato nelle operazioni morfologiche di chiusura (*imclose*). Questo coefficiente controlla la chiusura di piccoli gap, la fusione di regioni adiacenti e la regolarizzazione dei bordi. È impiegato esclusivamente quando l'*AreaScalingFactor* indica che l'oggetto è sufficientemente grande.

Successivamente all'inserimento dei parametri, viene inizializzato il modello preaddestrato *SAM* tramite la funzione *segmentAnythingModel*, dove l'oggetto restituito *samObj* rappresenta l'interfaccia principale per l'accesso alle funzionalità del modello, in particolare per l'estrazione degli embedding e la generazione delle maschere di segmentazione.

samObj = segmentAnythingModel;

Per ottimizzare le prestazioni computazionali, viene verificata la disponibilità di una GPU all'interno del sistema. Qualora questa sia disponibile, l'immagine è convertita in un array residente sulla GPU mediante la funzione *gpuArray*; in caso contrario, l'elaborazione avviene su CPU. Questa scelta consente di accelerare notevolmente le operazioni di deep learning nei casi in un l'hardware lo permetta.

Selezione iniziale guidata dall'utente

Una volta preparata l'immagine, sono estratti gli embedding del primo frame tramite la funzione *extractEmbeddings*; questi rappresentano una rappresentazione numerica ad alta dimensionalità dell'immagine, in cui ogni pixel è codificato in termini di caratteristiche semantiche, quali forma, texture e contenuto visivo. Questa rappresentazione è fondamentale, poiché costituisce la base su cui *SAM* applica i prompt per la segmentazione.

embeddingsFirst = extractEmbeddings(samObj, I_gpu);

Parallelamente, viene creata una finestra grafica *MATLAB* mediante la funzione *figure*, all'interno della quale sono definiti degli assi (*axes*) per la visualizzazione dell'immagine. L'immagine viene mostrata tramite *imshow*, specificando gli assi come contenitore. Inoltre l'handle dell'immagine (*hIm*) consente di aggiornare dinamicamente il contenuto visualizzato, permettendo la sovrapposizione delle maschere di segmentazione e il feedback visivo in tempo reale.

hIm = imshow(I, Parent=hAX) con *hAX = axes(f), f = figure;*

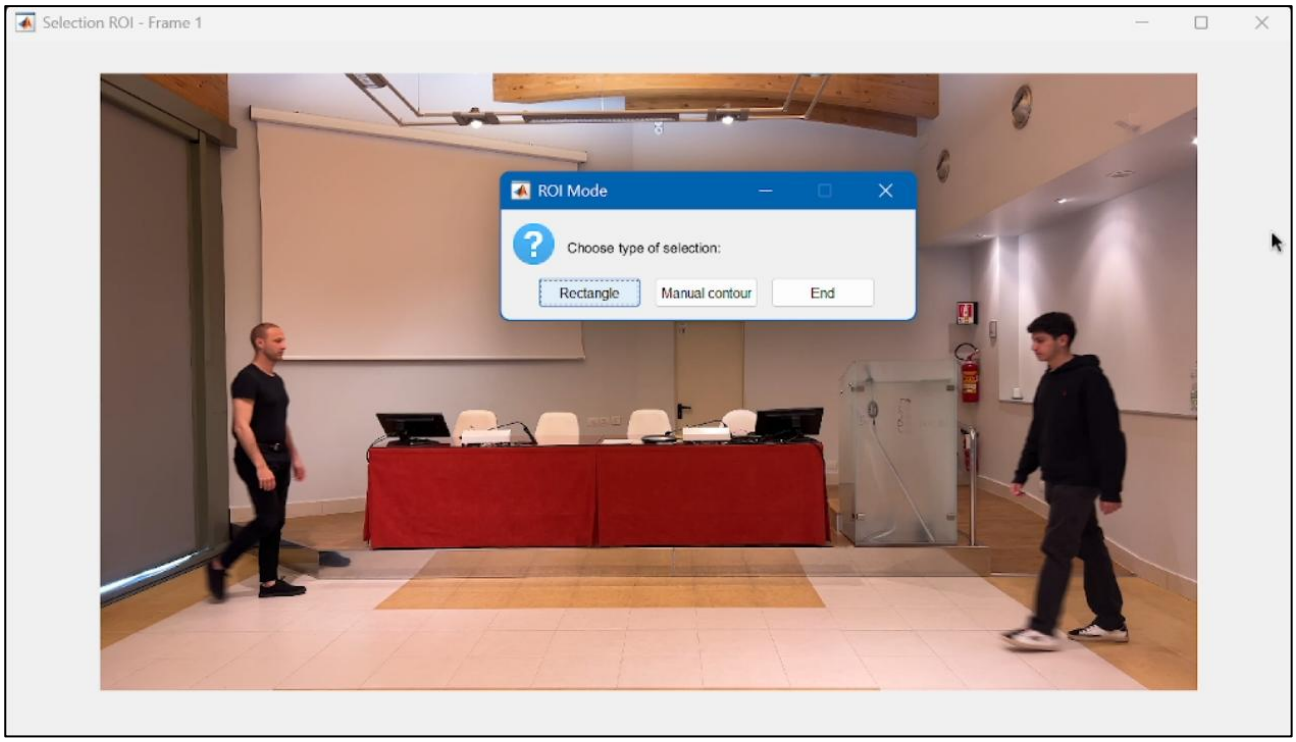

Figura 15: immagine corrispondente al primo fotogramma del video S2O2, con finestra di selezione della modalità di segmentazione.

Sul primo fotogramma, l'utente è chiamato a fornire una segmentazione iniziale, dove è data la possibilità di scegliere tra due modalità di selezione degli oggetti:

- Nella prima modalità, definita *manualContour*, l'utente definisce direttamente il contorno dell'oggetto d'interesse mediante una selezione libera a mano, producendo una maschera binaria esplicita. In questo caso non è utilizzata la segmentazione automatica di *SAM* per generare la maschera, difatti l'utente fornisce direttamente l'informazione spaziale completa.

- Per quanto riguarda la seconda modalità *Rectangle*, tramite la funzione *drawrectangle* l'utente può disegnare una o più *boundingbox* direttamente sull'immagine visualizzata. Ogni rettangolo disegnato non rappresenta la segmentazione finale, ma restituisce un oggetto ROI [32], dal quale vengono estratte posizione e dimensioni sotto forma di vettore.

roi = drawrectangle(hAX);

box = roi.Position; box = [x, y, larghezza, altezza];

Le informazioni sono passate a *SAM* come prompt spaziale mediante la funzione *segmentObjectsFromEmbeddings*, che utilizza gli embedding precedentemente calcolati, le

dimensioni dell'immagine e la bounding box per generare una maschera binaria dell'oggetto selezionato.

currMask = segmentObjectsFromEmbeddings(samObj, embeddingsFirst, size(I), BoundingBox=box)

Rispettivamente le due modalità di selezione presentano vantaggi e limitazioni:

- la modalità di selezione *manualContour* è particolarmente efficace per oggetti irregolari, strutture sottili oppure casi ambigui per il modello *SAM*, ma allo stesso tempo è limitata dal maggior tempo di selezione e dalla dipendenza delle abilità dell'utente;
- la modalità *Rectangle* garantisce una rapidità di selezione con minimo sforzo cognitivo per l'utente ed è altamente efficace in oggetti ben separati e con contorni netti. Al contempo possiede una minore precisione in presenza di oggetti adiacenti con forte sovrapposizione e nel caso di un background complesso.

Nel seguito sono riportate due immagini associate al primo fotogramma del filmato di videosorveglianza *S2O2*, dove è possibile distinguere con chiarezza la differenza tra le due modalità di selezione.

Figura 16: immagine corrispondente al primo fotogramma del video S2O2, selezione di un soggetto tramite modalità manualContour.

Figura 17: immagine corrispondente al primo fotogramma del video S2O2, selezione di un soggetto tramite modalità Rectangle.

Le maschere prodotte vengono successivamente accumulate in una maschera globale (*BW*) tramite un'operazione di unione logica, consentendo di combinare più selezioni dell'utente in un'unica rappresentazione del foreground. Per fornire un riscontro immediato all'utente delle selezioni compiute, la maschera viene sovrapposta visivamente all'immagine originale tramite la funzione *insertObjectMask*. L'immagine visualizzata viene quindi aggiornata modificando la proprietà *CData* dell'handle grafico, permettendo di osservare in tempo reale il risultato della selezione.

imgOverlay = insertObjectMask(gather(I), BW);
hIm.CData = imgOverlay;

Segmentazione automatica sui frame successivi

Una volta completata la selezione manuale, *SAM* viene inizializzato nuovamente come misura di controllo, al fine di evitare che eventuali stati interni alterati influenzino le elaborazioni successive. Sono quindi ricalcolati gli embedding del primo frame. Le dimensioni dell'immagine vengono memorizzate per garantire che tutte le maschere generate siano correttamente allineate ai frame originali. A partire dalla maschera binaria ottenuta, la funzione *regionprops* viene utilizzata per individuare tutte le regioni segmentate e calcolarne le rispettive bounding box minime. Le bounding

box estratte sono poi concatenate in una matrice, in modo da costituire i prompt iniziali per la segmentazione automatica dei fotogrammi successivi.

```
stats = regionprops(BW, 'BoundingBox');
bboxes = vertcat(stats.BoundingBox);
```

A partire dal secondo fotogramma, l'algoritmo entra nella fase di elaborazione automatica, con l'obiettivo di isolare in modo coerente gli stessi soggetti lungo l'intera sequenza video. A tal fine, sono inizializzate alcune variabili di stato che rappresentano il riferimento corrente del processo come l'immagine associata, gli embedding collegati e l'insieme delle bounding box corrispondenti agli oggetti segmentati nel frame precedente. Queste informazioni vengono aggiornate iterativamente per ciascun nuovo fotogramma, permettendo all'algoritmo di adattarsi dinamicamente alle variazioni temporali della scena.

Per ogni frame successivo al primo, l'immagine viene caricata e, se disponibile, trasferita sulla GPU per accelerare le operazioni di deep learning. In seguito, per ciascuna bounding box attiva, il modello *SAM* viene interrogato tramite la funzione *segmentObjectsFromEmbeddings* già richiamata nelle fasi inziali, utilizzando gli embedding correnti e le coordinate spaziali dell'oggetto come prompt. Questo approccio è particolarmente efficiente, in quanto consente di riutilizzare le informazioni semantiche già apprese, evitando una nuova fase di selezione.

```
for i = 1:size(currentBBoxes,1)
        currMask = segmentObjectsFromEmbeddings(...);
        totalMask = totalMask | gather(currMask);
end
```

La segmentazione viene eseguita oggetto per oggetto, generando una maschera binaria per ciascuna bounding box. Le maschere risultanti vengono quindi combinate mediante un'operazione di unione logica, producendo una maschera globale (*totalMask*) del foreground per il frame corrente. Tale maschera rappresenta l'insieme completo dei soggetti in movimento identificati nel fotogramma e costituisce l'output principale della fase di segmentazione automatica.

La strategia di segmentazione consente di mantenere una coerenza spaziale e semantica tra i frame, assicurando che gli oggetti inizialmente selezionati continuino ad essere tracciati e segmentati anche in presenza di lievi variazioni di posizione, forma o illuminazione.

Per fornire un'idea chiara riguardo le maschere elaborate dal modello *SAM* e al processo di segmentazione automatica appena descritto, nel seguito vengono mostrate due maschere associate rispettivamente al primo e ultimo frame del filmato di videosorveglianza *S2O2* acquisito durante l'esperienza sperimentale, in cui i soggetti selezionati sono identificati da pixel di colore bianco, mentre la restante parte associata al background è composta da pixel di colore nero.

Figura 18: immagine corrispondente alla maschera elaborata in modo semiautomatico sul primo fotogramma del video S2O2.

Figura 19: immagine corrispondente alla maschera elaborata in modo completamente automatico sull'ultimo fotogramma del video S2O2.

Modalità di correzione e raffinamento interattivo

Per rendere il processo di segmentazione robusto alle variazioni temporali e ridurre l'influenza dello sfondo, l'algoritmo introduce un meccanismo di aggiornamento dinamico degli embedding, strettamente integrato con un sistema di filtraggio del rumore. Una volta ottenuta la maschera globale del foreground per il frame corrente, viene generata una nuova versione dell'immagine in cui solo i pixel appartenenti agli oggetti segmentati sono preservati, mentre tutte le regioni esterne della maschera vengono azzerate.

L'immagine filtrata viene quindi utilizzata per ricalcolare gli embedding tramite *SAM*, in questo modo gli embedding del frame corrente risultano focalizzati unicamente sugli oggetti in primo piano, riducendo il contributo dello sfondo e migliorando la stabilità della segmentazione nei frame successivi. Parallelamente, dalla maschera globale sono estratte nuovamente le regioni connesse mediante la funzione *regionprops* già impiegata nei procedimenti precedenti, che consente di individuare le bounding box minime associate a ciascun oggetto segmentato.

Al fine di evitare la propagazione di artefatti nella segmentazione, viene applicato un filtro basato sull'area delle bounding box. In particolare, per ciascuna regione individuata viene calcolata l'area del rettangolo di delimitazione; le bounding box con area inferiore a una soglia prefissata (corrispondente al valore di *MinimumObjectSize* digitato dall'utente oppure pari a 100 pixel nel caso di oggetti non trovati nella maschera iniziale) vengono scartate, in quanto tipicamente riconducibili a rumore o regioni non significative. Solamente le bounding box che oltrepassano tale soglia vengono mantenute e utilizzate come prompt per la segmentazione del frame successivo.

$$Area_{bbox} = W \times H \geq A_{bbox,min}$$

Questo doppio meccanismo di aggiornamento degli embedding e filtraggio delle bounding box consente di aumentare la robustezza del tracciamento degli oggetti nel tempo, riducendo l'accumulo di errori tra fotogrammi consecutivi.

A questa procedura appena descritta, il codice introduce ulteriori operazioni morfologiche mirate al raffinamento delle maschere binarie, specialmente nei casi di imperfezioni strutturali, quali discontinuità nei contorni, fori interni, frammentazioni o regioni falsamente identificate. Tra queste operazioni si distinguono rispettivamente:

- La **chiusura morfologica** (*imclose*), che si pone l'obiettivo di migliorare la continuità e la compattezza delle regioni segmentate senza modificare in modo significativo la geometria globale degli oggetti. Consiste nell'applicazione sequenziale di due trasformazioni, rispettivamente la dilatazione (*dilation*) e l'erosione (*erosion*) [33], applicate allo stesso elemento strutturante. Formalmente, data una maschera binaria *A* e l'elemento strutturante *SE = strel('disk', r)* , la chiusura è definita come:

$$A \bullet SE = (A \oplus SE) \ominus SE$$

dove:

- $\oplus$ indica la dilatazione e $\ominus$ indica l'erosione;

La chiusura morfologica può essere descritta come segue:

- *dilation* espande leggermente le regioni segmentate, facendo crescere i bordi degli oggetti, per cui piccoli vuoti o interruzioni sono temporaneamente colmati;
- *erosion* riporta l'oggetto approssimativamente alla dimensione originale, di seguito le parti che erano state collegate durante la dilatazione restano unite, mentre l'espansione eccessiva viene rimossa.

Il risultato finale è una regione che mantiene la forma complessiva originale, con contorni più continui e con l'assenza di piccoli gap o discontinuità. È importante sottolineare che nel codice sviluppato, *imclose* viene applicata in modo condizionato, ovvero solo quando gli oggetti segmentati hanno dimensioni sufficientemente grandi.

- Il **riempimento delle cavità** (*imfill*), che ha lo scopo di trasformare una regione binaria in un'area topologicamente piena, eliminando buchi interni completamente circondanti da pixel di foreground.

Imfill(A, 'holes');

Formalmente, dato un insieme binario *A*, questa operazione morfologica produce una nuova maschera A' tale che:

$$A' = A \cup H$$

Dove H è l'insieme dei pixel appartenenti a cavità chiuse all'interno di A.

L'operazione è applicata dopo *imclose* e, consente di eliminare fori dovuti a errori di segmentazione, migliorare la coerenza geometrica degli oggetti e stabilizzare il calcolo di area e bounding box.

- La **rimozione delle componenti di piccola area** (*bwareaopen*), che ha come compito di rimuovere automaticamente tutte le componenti connesse la cui area è inferiore a una soglia prefissata, per cui:

$$A' = \bigcup_{Area(C_i) \geq A_{min}} C_i$$

dove C_i sono le componenti connesse alla maschera e A_{min} è la soglia minima di area. Nel codice questa soglia è direttamente legata al *MinimumObjectSize* ed è indirettamente adattata tramite lo *ScaleFactor*, che determina A_{min} in base alla media iniziale degli oggetti.

Questa operazione consente di eliminare il rumore residuo, rimuovere frammenti isolati e prevenire la formazione di falsi positivi. Rappresenta il filtro di pulizia fondamentale, garantendo che solo le regioni semanticamente rilevanti vengano propagate ai frame successivi.

Durante la fase di elaborazione ed estrazione delle maschere, l'utente è informato del progresso riguardo la segmentazione mediante la visualizzazione di un 'barra di progresso' (*progress bar*), tramite cui è in grado di interagire in ogni momento per terminare il processo oppure interromperlo temporaneamente per correggere le maschere calcolate in modo errato.

In quest'ultimo caso, il codice prevede un meccanismo di correzione interattivo guidato dall'utilizzatore, che consente di modificare o integrare le maschere esistenti e di rilanciare il processo di segmentazione. Questa strategia introduce un approccio *human-in-the-loop*, nel quale l'intervento umano viene utilizzato per riallineare il risultato automatico con l'intento semantico desiderato. Sono quindi forniti rispettivamente due procedimenti correttivi:

1. La modalità '*Overwrite All*' consente all'utente di sostituire completamente le maschere di segmentazione esistenti con una nuova segmentazione manuale. In questa configurazione, tutte le informazioni derivate dalle segmentazioni precedenti sono deliberatamente scartate, perciò l'utente effettua una nuova selezione dei soggetti d'interesse ed il modello viene rilanciato esclusivamente sulla base della nuova selezione. Con questo approccio è garantita la massima libertà di intervento all'utente con azzeramento della propagazione di errori precedenti e pieno riallineamento tra segmentazione e obiettivo applicativo.

2. La modalità '*Add to existing Masks*' permette all'utilizzatore di integrare la segmentazione corrente, aggiungendo nuove regioni di interesse senza eliminare le maschere già presenti. Tale procedura è progettata per effettuare correzioni locali e incrementali, mantenendo la coerenza con i risultati precedenti. È particolarmente indicata quando la segmentazione automatica è sostanzialmente corretta, ma sono presenti oggetti mancanti o parzialmente segmentati. A livello metodologico sono preservate le conoscenze già acquisite dal sistema, minimizzando l'impatto delle correzioni sull'intera scena. Questa modalità favorisce una convergenza graduale verso una segmentazione stabile a accurata.

Sulla base di quanto riportato, si evince come le diverse modalità di correzione rappresentino un elemento chiave per garantire l'accuratezza e l'affidabilità del processo di segmentazione. L'intervento da parte dell'utente consente di correggere imprecisioni generate automaticamente dal modello, migliorando progressivamente la qualità delle maschere prodotte. Questo approccio *human-in-the-loop* permette di integrare le elevate prestazioni di *SAM* con il controllo esperto umano, risultando particolarmente vantaggioso in scenari complessi o caratterizzati da ambiguità visive.

Per evidenziare l'applicazione dell'algoritmo nei diversi scenari, sono quindi mostrate quattro immagini corrispondenti rispettivamente alla selezione dei soggetti e alle maschere binarie elaborate sul primo frame nei filmati analisi macroscopica *S2A2* e di microscopia *Microorganism_Motility_01*.

Figura 20: immagine corrispondente al primo fotogramma del video S2A2, selezione di più soggetti tramite modalità Rectangle.

Figura 21: immagine corrispondente alla maschera elaborata in modo manuale sul primo fotogramma del video S2A2.

Figura 22: immagine corrispondente al primo fotogramma del video Microorganism_Motility_01, con selezione di più soggetti tramite modalità Rectangle.

Figura 23: immagine corrispondente alla maschera elaborata in modo semiautomatico sul primo fotogramma del video Microorganism_Motility_01.

6.3 Graphical User Interface (GUI)

Una GUI [34] è un'interfaccia utente basata sull'impiego di elementi grafici quali finestre, pulsanti, icone, menù e campioni di inserimento, che permettono all'utente di interagire con un sistema software in modo intuitivo e immediato, senza necessità di ricorrere a comandi testuali. Una GUI ben progettata contribuisce in maniera significativa a migliorare l'efficienza operativa e l'esperienza dell'utente, riducendo il margine di errore, aumentando la produttività e facilitando l'utilizzo anche da parte di utilizzatori non esperti. Nel contesto di questo lavoro, una volta completata la progettazione dell'algoritmo di *Foreground Detection*, si è ritenuto opportuno sviluppare un'interfaccia grafica dedicata, in grado di rendere l'interazione con l'algoritmo più semplice e controllata. La GUI consente infatti di gestire in modo agevole le fasi di caricamento dei dati, la configurazione dei parametri di segmentazione e l'avvio delle procedure di elaborazione, migliorando l'usabilità dell'applicazione.

Per la realizzazione della GUI è stato utilizzato *App Designer* di *MATLAB*, uno strumento che permette di progettare applicazioni interattive combinando un layout grafico organizzato con la logica di calcolo implementata in *MATLAB*.

App Designer

Consiste in un *ambiente di sviluppo integrato* (IDE) [35] per la creazione di interfacce grafiche utente, che consente di progettare applicazioni senza dover scrivere il codice per la disposizione degli elementi grafici. Esso si basa su un editor visuale *drag and drop* [36], attraverso il quale è possibile organizzare e inserire facilmente componenti grafici come pulsanti, pannelli, tabelle, caselle di testo, controlli numerici e aree di visualizzazione delle immagini. L'ambiente è strutturato in due sezioni principali:

- *Design View*, dedicata alla progettazione e disposizione degli elementi dell'interfaccia;
- *Code View*, nella quale sono definite le funzioni di callback e la logica applicativa associata ai vari componenti grafici.

Ogni componente dell'interfaccia può essere collegato a funzioni *MATLAB*, permettendo l'esecuzione di operazioni di calcolo, l'elaborazione di immagini o la gestione di flussi di dati in risposta alle azioni dell'utente. Le applicazioni sviluppate possono essere eseguite direttamente

all'interno di *MATLAB* oppure esportate come file in formato *.mlapp* o come applicazioni eseguibili standalone.

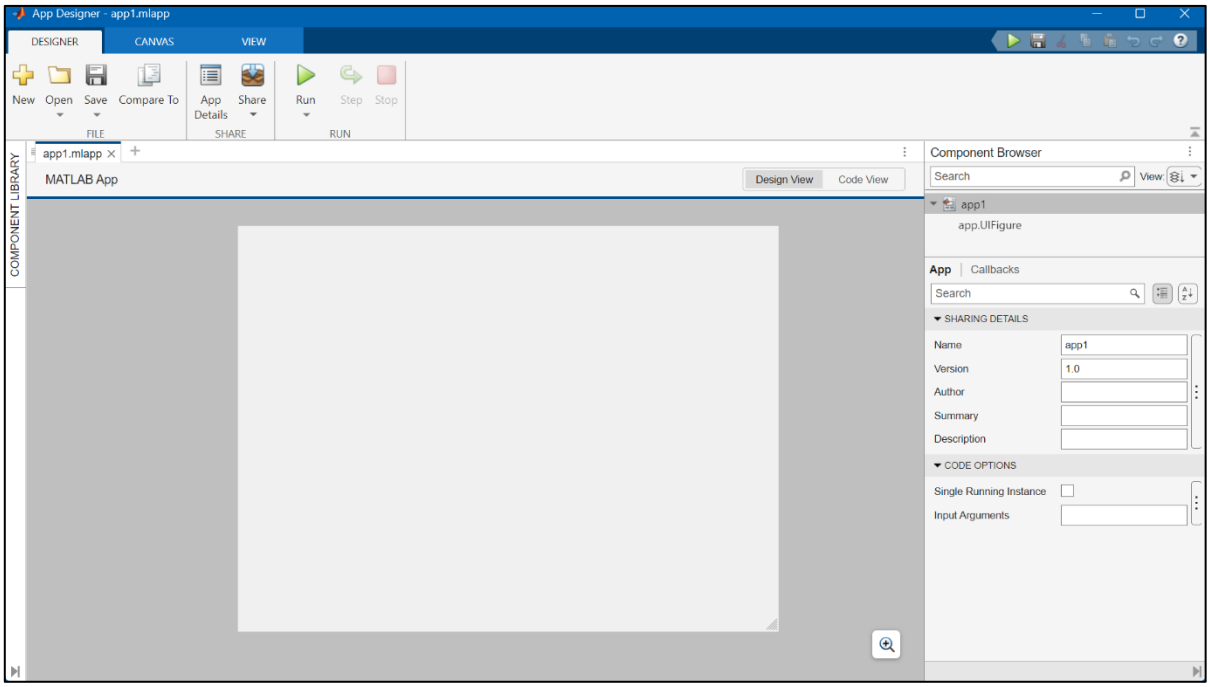

Figura 24: immagine corrispondente all'interfaccia grafica di AppDesigner, con visualizzata la sezione di Design View.

Interfaccia grafica per l'algoritmo di Foreground Detection

L'interfaccia grafica sviluppata per l'algoritmo di *Foreground Detection* è progettata per supportare l'utente durante tutte le fasi principali del processo di segmentazione. In particolare, essa consente di caricare i frame video, impostare i parametri necessari al funzionamento dell'algoritmo e salvare i risultati intermedi e finali.

L'applicazione permette di generare maschere binarie associate agli elementi del primo piano, che vengono successivamente applicate ai fotogrammi del video originale tramite la principale interfaccia di *ViFoSe*, al fine di isolare esclusivamente gli oggetti in movimento, escludendo lo sfondo.

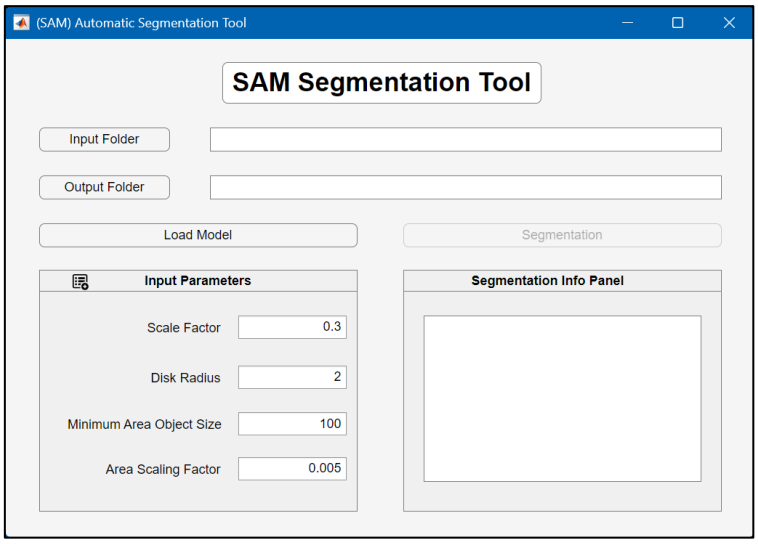

Figura 25: immagine corrispondente all'interfaccia App Designer del tool di Foreground Detection.

Per fornire maggiore chiarezza riguardo al meccanismo dell'interfaccia, sono indicate le diverse componenti che la costituiscono, specificando il funzionamento di ciascuna:

- Pulsante **'SAM Segmentation Tool'**: identifica chiaramente il titolo dell'applicazione, richiamando il nome del modello *SAM* utilizzato. Avviata l'applicazione e cliccato su tale pulsante, l'utente è in grado di visualizzare una serie di dettagli tecnici relativi al 'tool' ed i requisiti *MATLAB* per poterlo utilizzare. Questo elemento ha una funzione puramente informativa e contribuisce a migliorare la chiarezza e la riconoscibilità dello strumento.

- Pulsante '**Input Folder**': rappresenta il punto di ingresso dei dati, per cui l'utente può selezionare la cartella contenente i frame del video da elaborare. Tramite una finestra di dialogo è possibile indicare il percorso della directory di input, che viene visualizzato nel campo di testo adiacente. In seguito i file sono caricati e ordinati in modo numerico, con una successiva verifica da parte del sistema ed eventuale segnalazione di errori.

- Pulsante '**Output Folder**': permette di selezionare la directory di destinazione in cui sono salvate le maschere binarie generate dall'algoritmo. In modo analogo a prima, il percorso selezionato è mostrato nel campo di testo corrispondente.

- Pulsante '**Load Model**': ha il compito di inizializzare *SAM* all'interno dell'applicazione. Dopo la sua attivazione, viene mostrata una finestra di avanzamento che segnala il caricamento del modello con successiva creazione dell'oggetto *segmentAnythingModel*. Al termine del caricamento, è abilitato il pulsante di segmentazione.

- **Pannello** '**Input Parameters**': raccoglie tutti i parametri numerici che permettono di controllare il comportamento dell'algoritmo durante la segmentazione, sia nella fase automatica, sia nelle operazioni di post-processing. L'utente può digitare manualmente i valori corrispondenti a *Scale Factor*, *Disk Radius*, *Minimum Area Object Size* e *Area Scaling Factor* nelle corrispettive caselle di testo.

    - 'Icona di aiuto ai parametri': accanto al pannello dei parametri è presente un'icona informativa che, se selezionata, visualizza una finestra di supporto contenente una spiegazione dettagliata del significato e dell'effetto di ciascun parametro.

- Pulsante '**Segmentation**': avvia l'intero processo di *Foreground Detection* una volta che il modello è stato caricato e le cartelle iniziali sono state selezionate. Appena premuto, l'applicazione avvia l'interfaccia di selezione sul primo frame del video, richiedendo l'intervento da parte dell'utente. Definite le selezioni degli soggetti principali, l'algoritmo prosegue con la segmentazione automatica dei frame successivi.

- Pannello '**Segmentation Info Panel**': è dedicato alla visualizzazione delle informazioni di feedback calcolate durante l'esecuzione del codice. In particolare, sono mostrati l'area media degli oggetti segmentati, la dimensione minima degli oggetti calcolata automaticamente e messaggi di stato relativi alla segmentazione o alle eventuali correzioni. È fondamentale per fornire all'utente un riscontro quantitativo e immediato sull'andamento del processo.

Complessivamente, l'interfaccia grafica del *SAM Automatic Segmentation Tool* offre un equilibrio tra semplicità d'uso e controllo più approfondito dell'algoritmo. La suddivisone in componenti funzionali, unita alla possibilità di intervento manuale con visualizzazioni di feedback, rende la GUI uno strumento efficace per l'applicazione dell'algoritmo di *Foreground Detection* in numerosi contesti.

Interazione con l'interfaccia grafica *ViFoSe*

Il *SAM Automatic Segmentation Tool* rappresenta un modulo integrato all'interno dell'architettura complessiva del software *ViFoSe*, progettato per potenziare la fase di estrazione del primo piano dai contenuti video. In particolare, si colloca come uno degli stadi aggiuntivi al funzionamento

dell'interfaccia principale, con il compito di guidare l'utente nella generazione accurata delle maschere di segmentazione tramite *SAM*.

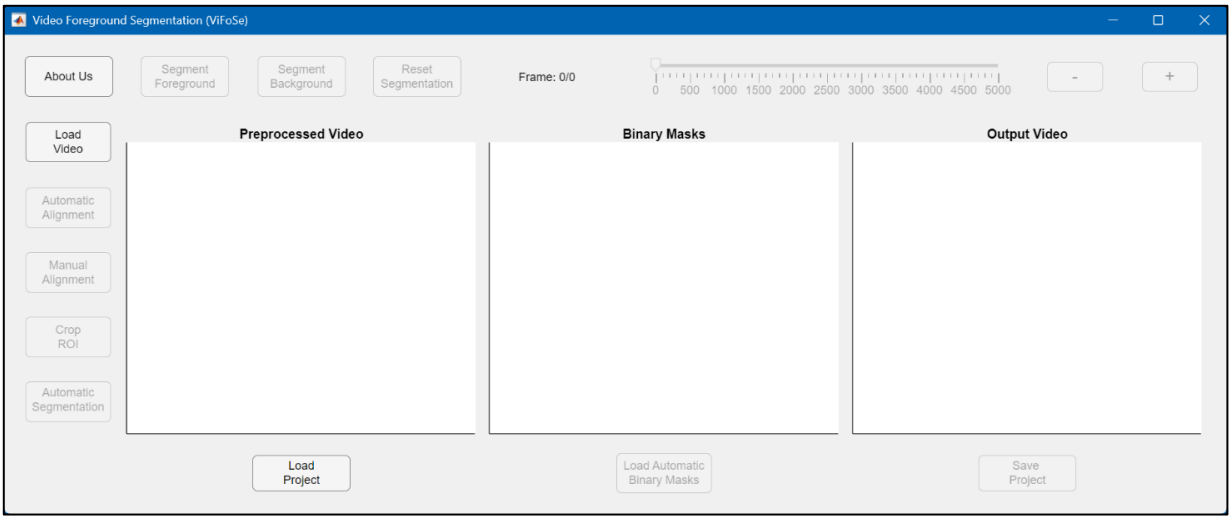

Figura 26: immagine corrispondente all'interfaccia ViFoSe a cui è integrato il tool di segmentazione semiautomatica.

Dal punto di vista operativo, il tool di segmentazione semiautomatica viene richiamato successivamente al caricamento del video originale all'interno di *ViFoSe*, dopo che il contenuto è stato convertito in una sequenza di frame e salvato sul file system. Qualora il filmato fosse caratterizzato da oscillazioni provate da una camera dinamica, il richiamo del tool può avvenire dopo l'applicazione delle procedure di stabilizzazione, garantendo che la segmentazione sia eseguita su dati spazialmente coerenti nel tempo.

In termini concreti, l'avvio dell'interfaccia del *SAM Automatic Segmentation Tool* avviene una volta premuto il pulsante apposito 'Automatic Segmentation', posizionato a sinistra nella *Design View* di *ViFoSe*.

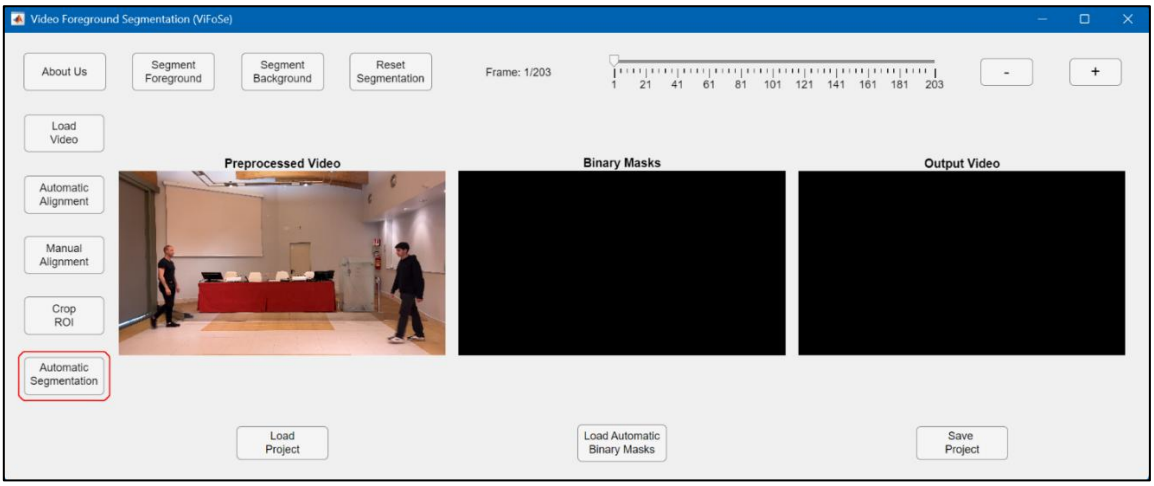

Figura 27: immagine corrispondente all'interfaccia ViFoSe con il video S2O2 caricato e il pulsante 'Automatic Segmentation' sbloccato.

Come specificato in precedenza, una volta completato il processo di generazione automatico delle maschere binarie tramite *SAM*, queste sono salvate in una directory di destinazione scelta dall'utente. Di conseguenza, l'integrazione con *ViFoSe* avviene per mezzo del pulsante 'Load Automatic Binary Masks' presente nell'interfaccia grafica principale. Mediante questa funzionalità l'utente può caricare direttamente le maschere salvate nella cartella di destinazione precedente all'interno dell'ambiente *ViFoSe*, rendendole disponibili per la visualizzazione e applicazione al video originale.

In questa maniera *ViFoSe* utilizza le maschere automatiche come base per la composizione dell'output finale, che include esclusivamente gli elementi del primo piano in movimento, escludendo lo sfondo. La stessa GUI può essere utilizzata come mezzo per raffinare e migliorare le segmentazioni create tramite *SAM*, così da ottenere un risultato più fine.

Nel complesso, il collegamento tra le due interfacce consente di integrare in modo efficace le capacità offerte da *SAM* all'interno del framework *ViFoSe*, mantenendo un flusso di lavoro guidato, e controllabile dall'utente.

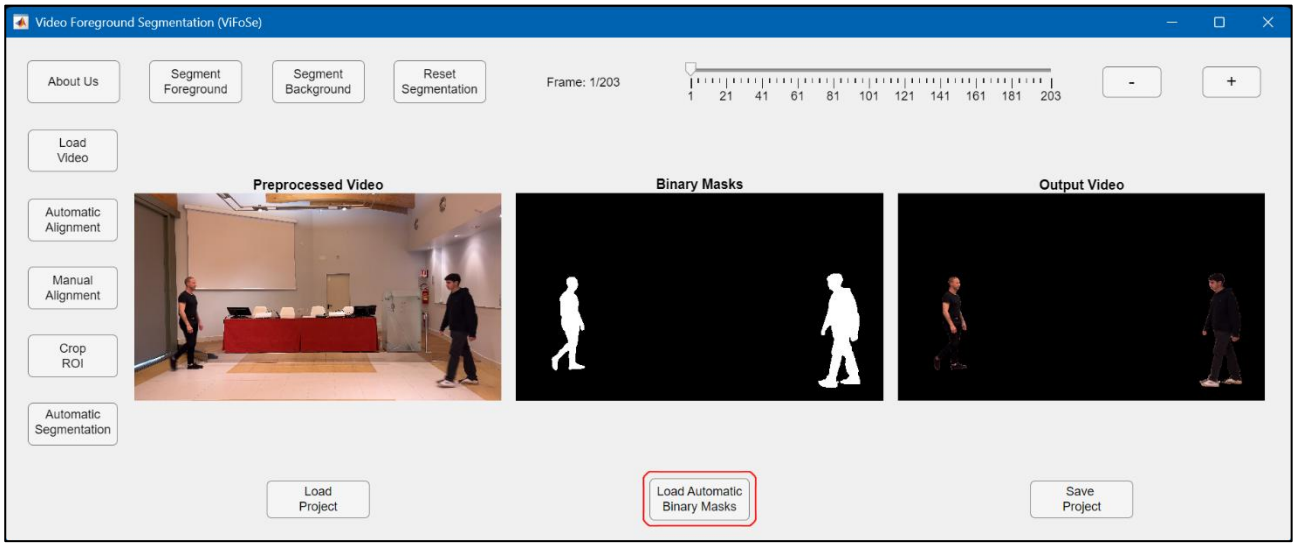

Figura 28: immagine corrispondente all'interfaccia ViFoSe con inserimento e applicazione delle maschere del video S2O2, caricate tramite il pulsante 'Load Automatic Binary Masks'.

7 Esperimenti

Il presente capitolo è dedicato alla descrizione e all'analisi della fase sperimentale, finalizzata alla valutazione delle prestazioni dell'algoritmo di *Foreground Detection* proposto nei diversi contesti applicativi. Per ciascun filmato sono illustrati i parametri quantitativi calcolati durante l'elaborazione, le impostazioni dei parametri di input definite dall'utente, nonché le principali criticità e limitazioni riscontrate nel processo di segmentazione.

7.1 Configurazione sperimentale

Questa sezione descrive il setup sperimentale adottato per l'esecuzione dei test, con particolare attenzione ai parametri definiti manualmente dall'utente e a quelli acquisiti durante l'utilizzo del tool. L'acquisizione dei diversi parametri è stata adottata facendo riferimento sia alle esigenze di confrontabilità dei risultati, sia ai principi consolidati della letteratura scientifica in ambito di segmentazione e analisi morfologica delle immagini [37] [38].

Parametri definiti manualmente

Per garantire una valutazione coerente e comparabile con le prestazioni dell'algoritmo, la configurazione sperimentale è stata mantenuta invariata per tutte le categorie di filmati considerati. In particolare, i parametri impostabili dall'utente tramite interfaccia grafica sono stati fissati agli stessi valori per ciascun esperimento, evitando influenze date da una regolazione manuale specifica per singolo dataset.

- Il *Scale Factor* è stato impostato con un valore pari a 0,3 consentendo una diminuzione del carico computazionale senza compromettere l'informazione spaziale necessaria alla segmentazione degli oggetti in movimento [39];

- Il *Disk Radius* è stato fissato a 2, tale da rimuovere i piccoli artefatti e migliorare la coerenza delle regioni segmentate senza alterare la struttura degli oggetti;

- La *Minimum Area Object Size* è stata stabilita a 100 pixel, come soglia per l'eliminazione delle componenti di area ridotta, riconducibili a rumore d'immagine. Questo valore è in grado di preservare gli oggetti reali presenti in molteplici contesti [40];

- L'*Area Scaling Factor* è stato configurato con un valore di 0,005 per normalizzare le operazioni dipendenti dall'area degli oggetti segmentati.

Parametri acquisiti durante la segmentazione

Oltre alla definizione dei parametri di input dell'algoritmo, la configurazione sperimentale ha previsto la raccolta di un insieme di parametri quantitativi finalizzati a caratterizzare il comportamento del processo di segmentazione dal punto di vista operativo e computazionale. Tali misure hanno l'obiettivo di valutare l'efficienza complessiva del tool sviluppato, quindi del metodo proposto.

Tra le diverse grandezze misurate si distinguono:

- Il *numero di frames segmentati manualmente*, corrispondenti ai fotogrammi segmentati tramite modalità *manualContour*, con intervento da parte dell'utente, tali da fornire un'indicazione del livello di complessità del contenuto visivo;

- Il *numero di frames segmentati in modalità semiautomatica*, associati ai fotogrammi segmentati in modalità *Rectangle*, consentendo al modello *SAM* di guidare la selezione delle regioni di interesse, con breve intervento da parte dell'utente;

- Il *numero di frames elaborati senza alcun intervento diretto*, coincidenti con i fotogrammi segmentati automaticamente dal modello a partire dalle informazioni precedenti. Tale parametro consente di quantificare la capacità dell'algoritmo di mantenere la coerenza della segmentazione nel tempo.

- Il *tempo di estrazione delle maschere*, legato all'intervallo di attesa in cui sono generate le maschere binarie associate a ciascun fotogramma, senza alcun tipo di interruzione.

- Il *tempo impiegato per la segmentazione manuale*, riferito alle operazioni di selezione tramite contorno libero sia sul primo frame sia durante le correzioni;

- Il *tempo impiegato per la segmentazione semiautomatica*, associato ai momenti in cui l'utente interviene tramite modalità *Rectangle* con selezioni soddisfacenti;

- Il *tempo di elaborazione successivo a interventi correttivi*, corrispondente all'intervallo di tempo necessario per aggiornare le maschere e propagare le modifiche ai fotogrammi successivi nel caso di segmentazioni errate. È fondamentale per valutare l'impatto delle correzioni sull'intero flusso di elaborazione.

- Il *tempo di elaborazione complessivo*, comprende tutte le fasi del processo di segmentazione, coincidente con la somma di tutte le 'forme temporali' precedentemente riportate. Esso fornisce una misura dell'efficienza globale del sistema, consentendo un diretto confronto tra le diverse categorie di filmati.

7.2 Risultati e problematiche riscontrate

Sulla base di quanto descritto in precedenza, sono quindi analizzati e riportati i valori ottenuti durante la fase sperimentale per ogni categoria di filmato, specificando per ciascuno di essi le principali problematiche riscontrate durante le fasi di segmentazione.

Filmato di videosorveglianza con un soggetto (i.e. *S1O1*)

Rappresenta il caso base ed è caratterizzato da condizioni di acquisizione favorevoli e da una scena relativamente semplice, per cui nel corso dell'intera sequenza non si sono verificate perdite di tracciamento del soggetto in movimento. È stata necessaria un'unica selezione in modalità semiautomatica *Rectangle*, effettuata esclusivamente sul primo frame, mentre non è stato effettuato alcun intervento in modalità *manualContour*. Il tracciamento è quindi risultato continuo e stabile per tutta la durata del filmato.

In termini di numero di fotogrammi, si sono riscontrati i rispettivi valori:

- *numero di frames segmentati manualmente* = 0;
- *numero di frames segmentati in modalità semiautomatica* = 1;
- *numero di frames elaborati senza alcun intervento diretto* = 139.

In relazione ai tempi di acquisizione, sono stati ottenuti i seguenti valori:

- *tempo di estrazione delle maschere* = 4569 s;
- *tempo impiegato per la segmentazione manuale* = 0 s;
- *tempo impiegato per la segmentazione semiautomatica* = 6,58 s;
- *tempo di elaborazione successivo a interventi correttivi* = 0 s;
- *tempo di elaborazione complessivo* = 4575,58 s.

Le principali imprecisioni riscontrate riguardano alcune fasi specifiche del movimento del soggetto, in particolare nei momenti in cui le gambe si avvicinano per riprendere il passo. In tali circostanze, lo spazio intermedio tra gli arti inferiori viene talvolta erroneamente incluso nella regione di foreground. Tali imprecisioni non compromettono in modo significativo la qualità finale della segmentazione, poiché il contorno del soggetto rimane sempre ben definito e privo di alterazioni rilevanti.

Filmato di videosorveglianza con due soggetti (i.e. *S2O2*)

Questo secondo filmato presenta una scena più complessa, caratterizzata dalla presenza simultanea di due soggetti in movimento e da frequenti situazioni di sovrapposizione e occlusione parziale, che hanno inciso in modo significativo sul processo di segmentazione.

In termini di numero di fotogrammi, si sono riscontrati i rispettivi valori:

- *numero di frames segmentati manualmente* = 8;
- *numero di frames segmentati in modalità semiautomatica* = 6;
- *numero di frames elaborati senza alcun intervento diretto* = 108.

In relazione ai tempi di acquisizione, sono stati ottenuti i seguenti valori:

- *tempo di estrazione delle maschere* = 12955 s;
- *tempo impiegato per la segmentazione manuale* = 167,39 s;
- *tempo impiegato per la segmentazione semiautomatica* = 235,68 s;
- *tempo di elaborazione successivo a interventi correttivi* = 2980 s;

- *tempo di elaborazione complessivo* = 16338,07 s.

Entrambi i soggetti sono stati selezionati in modalità semiautomatica a partire dal primo frame. Il primo dei due è stato segmentato correttamente e in modo continuo fino al termine del filmato, senza richiedere ulteriori interventi. Il secondo soggetto, invece, ha manifestato diverse perdite di tracciamento nel corso della sequenza. Quest'ultimo, risulta più distante dalla camera rispetto al primo ed è quindi caratterizzato da una sagoma di dimensioni più piccole, fattore che ha inciso negativamente sulla sua individuazione. Per un numero limitato di fotogrammi, pari a 6, i soggetti sono stati segmentati correttamente in modo completamente automatico senza necessità di intervento, con un tempo di elaborazione di circa 7 minuti e 38 secondi. Dopo questo intervallo si è verificata la prima perdita di tracciamento del secondo soggetto. Ulteriori interruzioni del monitoraggio sono state osservate a partire dal frame 66 fino al frame 145, principalmente a causa dell'avvicinamento e della sovrapposizione tra i due soggetti. Tali condizioni sono state seguite da un'interruzione del movimento per circa metà della durata del filmato, durante la quale il secondo soggetto è rimasto posizionato dietro il primo. Gli ultimi episodi di mancato rilevamento si sono verificati nelle fasi finali del video.

In modo analogo a quanto osservato per il filmato *S1O1*, alcuni dei difetti verificati sono riconducibili al riconoscimento errato del foreground nelle aree intermedie tra le gambe dei soggetti, fenomeno che si presenta durante specifiche fasi del passo.

A livello computazionale, il video *S2O2* risulta quello che ha richiesto il tempo di elaborazione complessivo maggiore tra tutti i casi analizzati.

Filmato di analisi macroscopica (i.e. *S2A2*)

Quest'ultimo filmato, riguardante il contesto dell'analisi macroscopica, è contraddistinto da figure degli elementi da segmentare particolarmente complesse e da ricorrenti situazioni di occlusione, sia parziale sia totale, in alcuni tratti della sequenza. L'insieme di tali elementi, ha permesso di identificare questo video come il caso più critico tra quelli analizzati, specialmente in termini di selezione dei soggetti.

In termini di numero di fotogrammi, si sono riscontrati i rispettivi valori:

- *numero di frames segmentati manualmente* = 17;
- *numero di frames segmentati in modalità semiautomatica* = 0;
- *numero di frames elaborati senza alcun intervento diretto* = 12.

In relazione ai tempi di acquisizione, sono stati ottenuti i seguenti valori:

- *tempo di estrazione delle maschere* = 7856,55 s;
- *tempo impiegato per la segmentazione manuale* = 663,07 s;
- *tempo impiegato per la segmentazione semiautomatica* = 0 s;
- *tempo di elaborazione successivo a interventi correttivi* = 3811,1 s;
- *tempo di elaborazione complessivo* = 12330,72 s.

Per l'intera durata della ripresa, sono state filmate due formiche in movimento continuo, la prima delle quali ha mantenuto nell'insieme una posizione pressoché invariata nel tempo. La seconda invece, è stata caratterizzata da molteplici offuscamenti dovuti alla presenza di piccole aste in legno posizionate all'interno della *Piastra di Petri* prima delle riprese. A causa dell'elevata difficoltà nell'individuazione semiautomatica dei soggetti, legata alla complessità dei contorni, la selezione sul primo frame è stata effettuata direttamente in modalità *manualContour*. Il tracciamento iniziale è risultato soddisfacente fino al frame 13, senza particolari problematiche. Successivamente, si sono manifestate importanti perdite di tracciamento che hanno reso necessaria una nuova selezione manuale, nello specifico a partire dai frame 74, 132 e 160. I fotogrammi indicati coincidono con istanti della ripresa in cui il secondo soggetto risulta parzialmente coperto, determinando una perdita di rilevazione e rendendo indispensabile l'intervento manuale per ristabilire la correttezza del tracciamento.

Questo caso di studio evidenzia i limiti del modello in presenza di frequenti occlusioni e di soggetti caratterizzati da geometrie complesse, con movimenti rapidi, condizioni per le quali l'automazione del processo di segmentazione risulta meno efficace e fortemente dipendente dall'intervento dell'operatore.

Filmato con microrganismi in movimento (i.e. *Microorganism_Motility_01*)

Tale sequenza video, individua il contesto di microscopia biologica, dove la presenza simultanea di molteplici microorganismi in movimento, insieme alle frequenti situazioni di sovrapposizione con background dinamico, costituiscono un contesto di sperimentazione differente rispetto i filmati precedentemente analizzati.

In termini di numero di fotogrammi, si sono riscontrati i rispettivi valori:

- *numero di frames segmentati manualmente* = 4;
- *numero di frames segmentati in modalità semiautomatica* = 3;
- *numero di frames elaborati senza alcun intervento diretto* = 106.

In relazione ai tempi di acquisizione, sono stati ottenuti i seguenti valori:

- *tempo di estrazione delle maschere* = 5940 s;
- *tempo impiegato per la segmentazione manuale* = 35,38 s;
- *tempo impiegato per la segmentazione semiautomatica* = 53,1 s;
- *tempo di elaborazione successivo a interventi correttivi* = 1674 s;
- *tempo di elaborazione complessivo* = 7702,48 s.

All'interno della sequenza sono stati selezionati 6 soggetti su un totale di 27, col fine di velocizzare il processo di segmentazione. Tutti gli elementi sono stati inizialmente individuati in modalità semiautomatica sul primo fotogramma, grazie alla presenza di contorni ben definiti e facilmente distinguibili dallo sfondo. Il tracciamento si è mantenuto stabile per la maggior parte della sequenza. Le principali problematiche si sono manifestate tra il frame 40 e il frame 44, a causa della sovrapposizione di uno dei microrganismi con elementi di rumore e variazioni di contrasto del background. In corrispondenza di tali fotogrammi è stata necessaria una nuova selezione in modalità *manualContour*, al fine di reintegrare le porzioni del soggetto perse durante l'estrazione delle maschere. Infine, oltre a questi interventi manuali, sono state effettuate soltanto due ulteriori selezioni in modalità semiautomatica.

Le criticità osservate sono riconducibili all'elevata velocità di spostamento di alcuni elementi presenti nella scena, dovuta alla natura del filmato, corrispondente ad un timelapse. Inoltre gli effetti di

sovrapposizione con strutture non rilevanti hanno temporaneamente offuscato diversi dei soggetti in primo piano. Nel complesso, le forme dei microorganismi risultano ben definite e non presentano alterazioni significative lungo la sequenza.

Al fine di fornire una visione complessiva dei risultati sperimentali ottenuti, sono quindi riportati i grafici riassuntivi relativi ai parametri analizzati per ciascuna categoria di filmato. Queste rappresentazioni consentono un confronto immediato tra i diversi casi di studio, in termini di numero di interventi richiesti e di tempi di elaborazione.

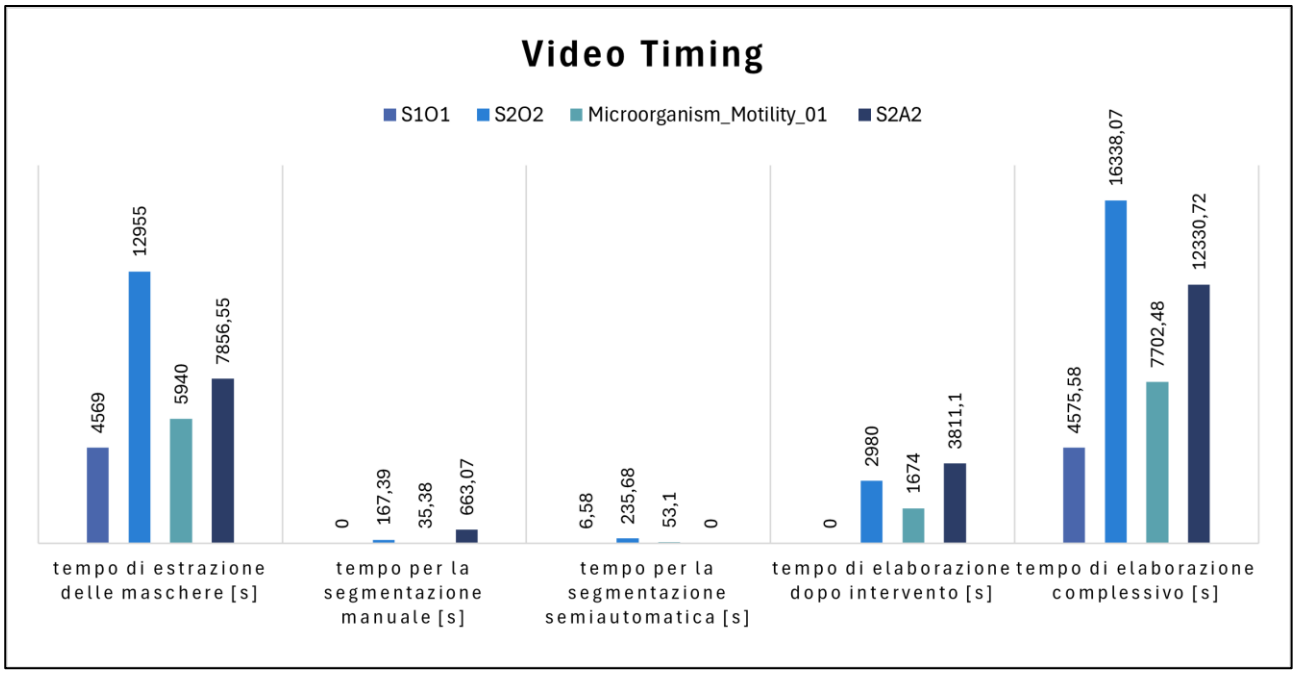

Figura 29: immagine corrispondente al grafico contenente le tempistiche sperimentali per ogni filmato.

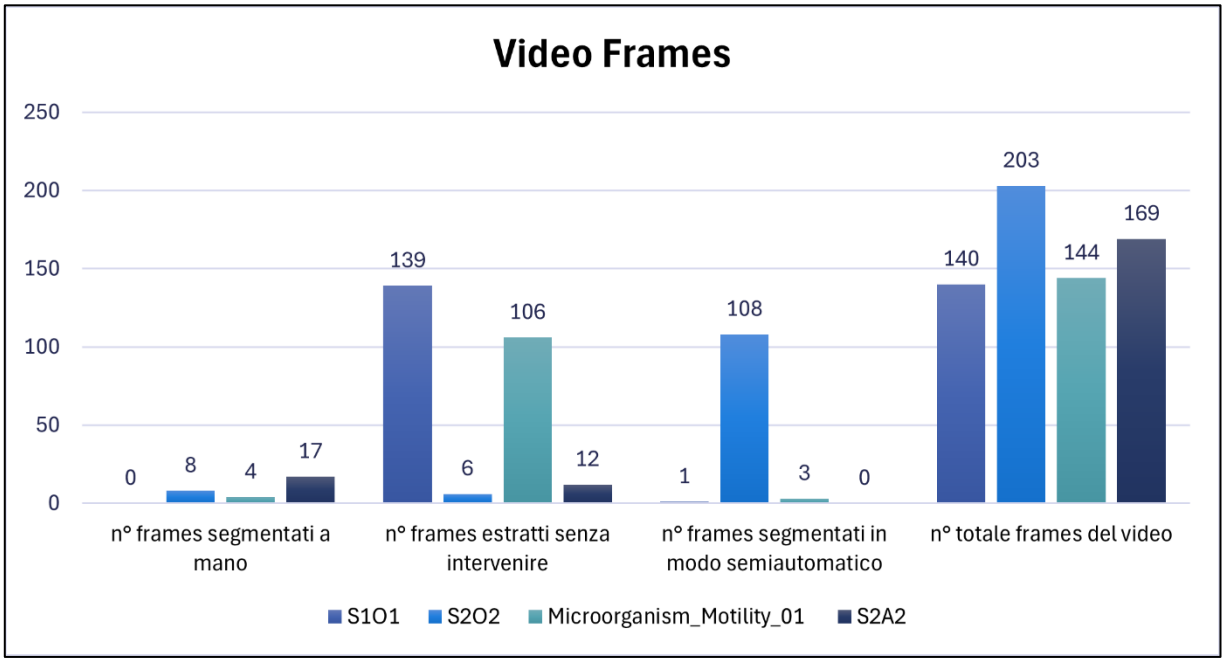

Figura 30: immagine corrispondente al grafico contenente i fotogrammi di elaborazione sperimentale per ogni filmato.

9 Conclusioni e sviluppi futuri

In questo lavoro abbiamo dimostrato come l'unione di algoritmi automatici di segmentazione con possibilità di correzione manuale sia in grado di fornire risultati in termini di segmentazione in numerose tipologie di filmati, consentendo una riduzione del numero di interventi manuali richiesti all'utente. In particolare, nei contesti caratterizzati da condizioni di acquisizione favorevoli e da scene relativamente semplici, l'algoritmo ha garantito un tracciamento stabile e continuo dei soggetti in movimento, con una buona qualità delle maschere generate. Nei casi più complessi, contraddistinti da geometrie articolate, frequenti sovrapposizioni e sfondi dinamici con grandi quantità di elementi, si è invece resa necessaria una maggiore interazione da parte dell'operatore, confermando la sua utilità nei processi di segmentazione.

I metodi di segmentazione implementati in questo studio hanno portato ad un'analisi esaustiva delle principali problematiche legate all'estrazione del primo piano in sequenze video con camera fissa, in presenza di sfondi statici e dinamici. L'impiego di maschere binarie, unite a operazioni di post-processing basate su tecniche di imaging morfologico, ha permesso di migliorare la qualità della segmentazione. L'applicazione delle maschere raffinate sui fotogrammi originali ha inoltre consentito di generare sequenze video finali, nelle quali lo sfondo risulta efficacemente rimosso, preservando gli elementi dinamici della scena che sono stati selezionati.

Un ulteriore aspetto indispensabile è rappresentato dall'integrazione del *SAM Automatic Segmentation Tool* all'interno dell'interfaccia grafica *ViFoSe*, che ha permesso di collocare l'algoritmo in un flusso di lavoro modulare. Tale aspetto, consente il riutilizzo delle maschere generate per successive fasi di analisi del movimento, aumentando la versatilità del sistema e rendendolo applicabile in vari contesti sperimentali.

Nonostante le prestazioni soddisfacenti, sono emerse alcune limitazioni intrinseche all'approccio adottato. In particolare, in presenza di soggetti caratterizzati da forme particolarmente complesse o da strutture sottili e articolate, il raffinamento dei contorni non risulta sempre completo. Inoltre, scenari con frequenti sovrapposizioni tra elementi dinamici evidenziano difficoltà nel mantenere un tracciamento corretto e continuo nel tempo. A tali problematiche si aggiunge anche l'occupazione delle matrici GPU, che insieme al mancato controllo in memoria, sono la causa di processi di elaborazione particolarmente lunghi e possibili crash del programma nel caso di dataset con grandi

dimensioni. L'insieme di queste criticità conferma la possibilità di introdurre miglioramenti nell'algoritmo e rappresenta un punto di partenza per futuri sviluppi di esso.

Lavori futuri

Sulla base delle limitazioni emerse è possibile individuare diverse direzioni di sviluppo finalizzate al miglioramento delle prestazioni e della robustezza del sistema.

1. **Ottimizzazione nella gestione della memoria e delle prestazioni computazionali**: in questo primo ambito, l'introduzione di meccanismi di controllo della memoria GPU supportato da strategie di monitoraggio dinamico, può ridurre il rischio di saturazione durante l'elaborazione di video di grandi dimensioni. Allo stesso tempo, l'acquisizione di tecniche di *parallel computing* [41], supportate dall'ambiente *MATLAB*, permette di ridurre i tempi di esecuzione, specialmente nel caso di numerosi oggetti da segmentare. Ulteriori sviluppi possono riguardare l'implementazione di meccanismi di 'recovery' automatici basati su sistemi di feedback, con l'obiettivo di riattivare procedure correttive in caso di errore, riducendo la necessità di interventi manuali. In tale direzione, un potenziamento nelle modalità di tracciamento mediante l'integrazione di algoritmi di tracking multi-oggetto, quali *SORT* [42] , *Deep SORT* [43] o *ByteTrack* [44], consente di preservare l'identità degli oggetti anche in presenza di occlusioni.

2. **Miglioramento della qualità della segmentazione ed estensione dell'algoritmo**: in questo secondo e ultimo ambito, l'accuratezza dei contorni può essere migliorata impiegando filtri sensibili (*edge aware filters*) [45], efficaci nel preservare le discontinuità in presenza di rumore. Anche tecniche come *Active Contours* o *Level Set Methods* possono essere utilizzate in fase di post-processing per migliorare la precisione delle maschere. L'aggiunta di operatori morfologici con *kernel adattivi* [46], la cui dimensione è regolata in funzione della scala degli oggetti segmentati, può bilanciare la rimozione del rumore ed il mantenimento dei dettagli. Per quanto riguarda scenari con la presenza simultanea di più oggetti, questi possono essere gestiti tramite approcci basati su modelli come *Mask R-CNN* [47] o *YOLACT* [48]. Infine, in ambito di imaging medicale, l'integrazione di modelli specialistici come il *Medical Segment Anything Model* (*MedSAM*) [49], unita a procedure di *fine-tuning* [50] fondate su dataset clinici dedicati, rappresenta una prospettiva importante per estendere il tool sviluppato a contesti diagnostici in cui è richiesta una maggiore accuratezza e affidabilità.

Nel complesso, il risultato raggiunto evidenzia come l'utilizzo di *foundation model* per la visione artificiale, opportunamente integrati in pipeline interattive, rappresenti una soluzione efficace per affrontare problemi di segmentazione complessi, mantenendo un buon compromesso tra automazione e controllo umano.

Il presente studio fornisce quindi una base solida per la segmentazione video e l'estrazione del primo piano, offrendo soluzioni pratiche applicabili sia a scenari statici sia a contesti più articolati. L'integrazione del tool di segmentazione basato su *SAM* all'interno di *ViFoSe*, reso disponibile pubblicamente nel repository *GitHub* dedicato (https://github.com/UniBoDS4H/ViFoSe), contribuisce a valorizzare ulteriormente il lavoro svolto, offrendo un ambiente software completo. All'interno del repository sono infatti disponibili il codice sorgente, un manuale d'uso dettagliato e un video tutorial che illustrano le modalità di utilizzo del software. Inoltre, al fine di garantire la riproducibilità degli esperimenti condotti, i filmati utilizzati nella fase sperimentale sono stati raccolti in un archivio di riferimento, il quale è scaricabile tramite il link *Google Drive* apposito (https://drive.google.com/file/d/1ZtG7WUmnp2bLoikkZCEZTBtr1_S_hDdf/view?usp=sharing).

Alla luce delle limitazioni mostrate e delle possibili direzioni di sviluppo individuate, l'algoritmo potrà essere ulteriormente potenziato per gestire scenari più complessi, ampliando il suo ambito di applicazione. In questo senso, il progetto presentato non si configura come un risultato conclusivo, ma come un punto di partenza per lavori futuri e approfondimenti nel campo della segmentazione video assistita da modelli di base.

7 Bibliografia

[1] Bouwmans, T., Silva, C., Marghes, C., Zitouni, M. S., Bhaskar, H., & Frelicot, C. (2018). On the role and the importance of features for background modeling and foreground detection. Computer Science Review, 28, 26-91.

[2] Butler, D. E., Bove, V. M., & Sridharan, S. (2005). Real-time adaptive foreground/background segmentation. EURASIP Journal on Advances in Signal Processing, 2005, 1-13.

[3] Zhang, Z., Jing, T., Ding, B., Gao, M., & Li, X. (2019). A model-based approach to foreground region of interest detection for video codecs. *Applied Sciences*, 9(13), 2670.

[4] Zhong, D., & Chang, S. F. (1999). An integrated approach for content-based video object segmentation and retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8), 1259-1268.

[5] Zhang, Z., Jing, T., Ding, B., Gao, M., & Li, X. (2019). A model-based approach to foreground region of interest detection for video codecs. Applied Sciences, 9(13), 2670.

[6] Kalsotra, R., & Arora, S. (2022). Background subtraction for moving object detection: explorations of recent developments and challenges. *The Visual Computer*, 38(12), 4151-4178.

[7] Fradi, H., & Dugelay, J. L. (2012, May). Robust foreground segmentation using an improved Gaussian mixture model and optical flow. *In proceedings of the 2012 IEEE International Conference on Informatics, Electronics & Vision (ICIEV)* (pp. 248-253).

[8] Lee, J., & Park, M. (2012). An adaptive background subtraction method based on kernel density estimation. *Sensors*, 12(9), 12279-12300.

[9] Zhang, Y., Ma, W., Yang, L., Duan, L., & Liu, J. (2014, November). Graph-cut-based interactive image segmentation with texture constraints. *In proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry* (pp. 57-63).

[10] Ilunga-Mbuyamba, E., Avina-Cervantes, J. G., Garcia-Perez, A., de Jesus Romero-Troncoso, R., Aguirre-Ramos, H., Cruz-Aceves, I., & Chalopin, C. (2017). Localized active contour model with background intensity compensation applied on automatic MR brain tumor segmentation. *Neurocomputing*, 220, 84-97.

[11] Harper, P., & Reilly, R. B. (2000, September). Color-based video segmentation using level sets. *In proceedings of the 2000 IEEE International Conference on Image Processing* (pp. 480-483).

[12] Du, G., Cao, X., Liang, J., Chen, X., & Zhan, Y. (2020). Medical image segmentation based on U-net: A review. *Journal of Imaging Science & Technology*, *64*(2).

[13] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

[14] Dai, C., Zheng, Y., & Li, X. (2006). Accurate video alignment using phase correlation. *IEEE Signal Processing Letters*, *13*(12), 737-740.

[15] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... & Girshick, R. (2023). Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 4015-4026).

[16] Asha Paul, M. K., Kavitha, J., & Jansi Rani, P. A. (2018). Key-frame extraction techniques: A review. Recent Patents on Computer Science, 11(1), 3-16.

[17] Al-Amri, S. S., & Kalyankar, N. V. (2010). Image segmentation by using threshold techniques. arXiv preprint arXiv:1005.4020.

[18] Mai, L., Le, H., Niu, Y., & Liu, F. (2011, December). Rule of thirds detection from photograph. In *2011 IEEE international symposium on multimedia* (pp. 91-96). IEEE.

[19] Lindeberg, T. (2012). Scale invariant feature transform.

[20] Bay, H., Tuytelaars, T., & Van Gool, L. (2006, May). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404-417). Berlin, Heidelberg: Springer Berlin Heidelberg.

[21] Gruosso, M., Capece, N., & Erra, U. (2021). Human segmentation in surveillance video with deep learning. *Multimedia Tools and Applications*, *80*(1), 1175-1199.

[22] Bertels, J., Eelbode, T., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., & Blaschko, M. B. (2019, October). Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice. In *International conference on medical image computing and computer-assisted intervention* (pp. 92-100). Cham: Springer International Publishing.

[23] Karimi, D., & Salcudean, S. E. (2019). Reducing the hausdorff distance in medical image segmentation with convolutional neural networks. *IEEE Transactions on medical imaging*, *39*(2), 499-513.

[24] Bertels, J., Eelbode, T., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., & Blaschko, M. B. (2019, October). Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice. In *International conference on medical image computing and computer-assisted intervention* (pp. 92-100). Cham: Springer International Publishing.

[25] Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. *ACM computing surveys (CSUR)*, *54*(10s), 1-41.

[26] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, *33*(12), 6999-7019.

[27] Perez, A. (2021, February). GeST: A New Image Segmentation Technique based on Graph Embedding. In *VISIGRAPP (4: VISAPP)* (pp. 245-252).

[28] Lempitsky, V., Kohli, P., Rother, C., & Sharp, T. (2009, September). Image segmentation with a bounding box prior. In *2009 IEEE 12th international conference on computer vision* (pp. 277-284). IEEE.

[29] Li, Y., Liang, F., Zhao, L., Cui, Y., Ouyang, W., Shao, J., ... & Yan, J. (2021). Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*.

[30] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 658-666).

[31] Zhang, Y., Zhou, T., Wang, S., Liang, P., Zhang, Y., & Chen, D. Z. (2023, October). Input augmentation with sam: Boosting medical image segmentation with segmentation foundation model. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 129-139). Cham: Springer Nature Switzerland.

[32] Lin, H., Si, J., & Abousleman, G. P. (2007, April). Region-of-interest detection and its application to image segmentation and compression. In *2007 International conference on integration of knowledge intensive multi-agent systems* (pp. 306-311). IEEE.

[33] Said, K. A. M., & Jambek, A. B. (2021, October). Analysis of image processing using morphological erosion and dilation. In *Journal of Physics: Conference Series* (Vol. 2071, No. 1, p. 012033). IOP Publishing.

[34] Jansen, B. J. (1998). The graphical user interface. *ACM SIGCHI Bulletin*, *30*(2), 22-26.

[35] Muşlu, K., Brun, Y., Holmes, R., Ernst, M. D., & Notkin, D. (2012). Speculative analysis of integrated development environment recommendations. *ACM SIGPLAN Notices*, *47*(10), 669-682.

[36] Jia, J., Sun, J., Tang, C. K., & Shum, H. Y. (2006). Drag-and-drop pasting. *ACM Transactions on graphics (TOG)*, *25*(3), 631-637.

[37] Serra, J. (1983). *Image analysis and mathematical morphology*. Academic Press, Inc..

[38] Dougherty, E. R. (1992). An introduction to morphological image processing. In *SPIE*. Optical Engineering Press.

[39] Lindeberg, T. (2009). Scale-space.

[40] Chan, A. H., & Courtney, A. J. (2000). Effects of object size and inter-object spacing on peripheral object detection. *International Journal of Industrial Ergonomics*, *25*(4), 359-366.

[41] Skillicorn, D. B., & Talia, D. (1998). Models and languages for parallel computation. *Acm Computing Surveys (Csur)*, *30*(2), 123-169.

[42] Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016, September). Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)* (pp. 3464-3468). Ieee.

[43] Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)* (pp. 3645-3649). IEEE.

[44] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., ... & Wang, X. (2022, October). Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision* (pp. 1-21). Cham: Springer Nature Switzerland.

[45] Gastal, E. S., & Oliveira, M. M. (2011). Domain transform for edge-aware image and video processing. In *ACM SIGGRAPH 2011 papers* (pp. 1-12).

[46] Van Kerm, P. (2003). Adaptive kernel density estimation. *The Stata Journal*, *3*(2), 148-156.

[47] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

[48] Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9157-9166).

[49] Zhang, Y., Shen, Z., & Jiao, R. (2024). Segment anything model for medical image segmentation: Current applications and future directions. *Computers in Biology and Medicine*, *171*, 108238.

[50] Friederich, S. (2017). Fine-tuning. *The Stanford encyclopedia of philosophy*.

Ringraziamenti

Desidero ringraziare le diverse persone che mi hanno accompagnato durante la realizzazione di questo progetto di Tesi.

In primo luogo, vorrei ringraziare il Prof. Emanuele Domenico Giordano dell'Università di Bologna, che mi ha seguito nel ruolo di Relatore durante la preparazione e lo svolgimento di questa Tesi, per avermi dato l'opportunità di lavorare a questo progetto multidisciplinare e di poter contribuire a un ambito molto stimolante come quello della "*Microscopy and Artificial Intelligence"*.

Desidero inoltre ringraziare il Prof. Filippo Piccinini, ricercatore presso IRST di Meldola, il Prof. Gastone Castellani, ordinario di Fisica dell'Università di Bologna, ed il Dott. Nicola Normanno per aver reso possibile questo progetto fornendo accesso ad immagini provenienti da varie fonti.

Un ringraziamento speciale va al Prof. Filippo Piccinini per il suo ruolo di correlatore in questa Tesi. La sua costante presenza e disponibilità, i suoi preziosi suggerimenti e il continuo supporto sono stati essenziali per raggiungere gli obiettivi che ci eravamo preposti. Ha condiviso in maniera chiara le sue conoscenze nel campo della microscopia e dell'analisi video, rendendo più comprensibili argomenti e procedure complessi. La sua gentilezza, competenza e professionalità hanno contribuito ad arricchire non solo questo progetto di Tesi, ma anche me come studente e come persona, e per questo gliene sono grato.

Infine vorrei dedicare un ringraziamento finale e particolarmente importante a tutte le persone che mi sono state vicine in questi anni.

Ringrazio la mia famiglia che prima di tutto e tutti mi ha accompagnato nel percorso universitario supportandomi anche nelle scelte più complicate.

Ringrazio la mia fidanzata e gli amici che ci sono stati nei momenti più difficili, dove sarebbe stato più semplice fermarsi piuttosto che lottare e andare avanti.