UNIVERSITÀ
DI PAVIA

FACOLTÀ DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN BIOINGEGNERIA

TESI DI LAUREA

TITOLO

SVILUPPO DI UN SISTEMA DI TRACCIAMENTO AUTOMATICO DI STRUMENTI LAPAROSCOPICI PER L'ANALISI DELLE PERFORMANCE CHIRURGICHE

Candidato: Eleonora Capra

Relatore: Prof.ssa Stefania Marconi
Correlatori: Dott.ssa Marta Botti
Prof.ssa Sara Condino
Prof. Vincenzo Ferrari

A.A. 2024/2025

Sommario

Indice delle figure 5

1. Introduzione 8

1.1. Stato dell'arte 11

1.2. Metodologia proposta 13

1.3. Metriche di interesse 15

1.4. Colecistectomia laparoscopica 17

1.4.1. Anatomia della colecisti 17

1.4.2. Procedura di rimozione della colecisti 20

2. Annotazione degli strumenti chirurgici mediante Roboflow 25

2.1. La piattaforma Roboflow 27

2.2. Creazione del dataset 29

2.2.1. Origine delle immagini 29

2.2.2. Definizione delle classi 31

2.2.3. Creazione delle label e delle bounding box 33

2.2.4. Data augmentation 35

2.2.5. Revisione delle annotazioni ed esportazione dei dati 37

3. Addestramento delle reti YOLO 40

3.1. Fondamenti dell'object detection 41

3.2. Evoluzione dei modelli YOLO e scelta dell'architettura 44

3.3. Architettura delle reti e implementazione dei modelli YOLOv8 e YOLOv26…… 47

3.3.1. Backbone 48

3.3.2. Neck 52

3.3.3. Detection head 55

3.4. Predizioni della rete ... 58
3.4.1. Bounding box ... 59
3.4.2. Confidence score ... 60
3.4.3. Class probability ... 61
3.4.4. Loss function ... 62
3.5. Implementazione sulla piattaforma Ultralytics ... 63
3.6. Addestramento delle reti e scelta della rete ottima ... 65
3.6.1. Configurazione degli iperparametri ... 66
3.6.2. Configurazione del modello ... 68
3.6.3. Andamento del training e analisi della funzione di loss ... 69
3.6.4. Metriche di valutazione del modello ... 72
4. Multi-Object Tracking (MOT) ... 79
4.1. BoT-SORT ... 81
4.2. ByteTrack ... 83
4.3. Scelta dell'algoritmo di tracking ... 84
5. Analisi qualitativa del tracking multi-oggetto ... 86
6. Post-processing dei dati e calcolo delle metriche ... 88
7. Discussione dei risultati ... 93
7.1. Fase di esposizione ... 93
7.2. Fase di isolamento ... 96
7.3. Fase di sezione ... 99
7.4. Fase di scollamento ... 102
7.5. Sviluppi futuri ... 104
7.6. Conclusione ... 107
Bibliografia ... 109

Indice delle figure

Figura 1.1 Tracking strumenti laparoscopici 10
Figura 1.2 Anatomia della colecisti 17
Figura 1.3 Triangolo di Calot 18
Figura 1.4 Posizionamento dei trocar per l'intervento di colecistectomia laparoscopica 21
Figura 1.5 Campo chirurgico in vista anteriore/mediale (A) e vista posteriore/laterale (B) del triangolo epatocistico e delle aree adiacenti. CBD: dotto biliare comune; Du. duodeno; AH: arteria epatica; RS: solco di Rouvière; Sg4: segmento 4; UF: fessura ombelicale……...….22
Figura 1.6 (A) Il triangolo di Calot viene dissezionato per esporre e delimitare il dotto cistico e l'arteria cistica. Il piano cistico non è ancora esposto (linea rotta). (B)Il piano cistico viene esposto. (C) La CP è esposta adeguatamente (linea interrotta). La punta bianca indica il dotto cistico e la punta nera indica l'arteria cistica. PC, placca cistica……...23
Figura 1.7 Applicazione delle clip sul dotto cistico e sull'arteria cistica durante colecistectomia laparoscopica 23
Figura 1.8 Dissezione della colecisti dal letto epatico durante colecistectomia laparoscopica. 24
Figura 2.1 Interfaccia della piattaforma Roboflow 28
Figura 2.2 Classi strumenti laparoscopici: (a) clipper, (b) grasper, (c) hook, (d) irrigator, (e) scissor 31
Figura 2.3 Difficoltà di visualizzazione dello strumento chirurgico: (a) frame con scarsa luminosità e obiettivo sporco, (b) frame con strumento parzialmente occluso da tessuto grasso 32
Figura 2.4 Creazione label e bounding box 34
Figura 2.5 Numero di etichette rilevate per classe 36
Figura 2.6 Divisione del dataset in training, validation e test set 39
Figura 3.1 Evoluzione delle reti YOLO dal 2015 al 2025 46
Figura 3.2 Architettura generica di una rete YOLO 47

Figura 3.3 Schema del blocco Conv–BN–Activation: sequenza di convoluzione, normalizzazione e funzione di attivazione utilizzata nelle reti neurali convoluzionali…….48
Figura 3.4 Architettura di una CSPnet..... 50
Figura 3.5 Schema della Feature Pyramid Network (FPN)..... 53
Figura 3.6 Implementazione delle predizioni nella detection head..... 55
Figura 3.7 Schema della detection head decoupled..... 56
Figura 3.8 Algoritmo di Non-Max-Suppression..... 57
Figura 3.9 Creazione delle bounding boxes..... 59
Figura 3.10 Algoritmo di Intersection over Union (IoU)..... 61
Figura 3.11 Scelta degli iperparmetri di training..... 67
Figura 3.12 Valori finali delle principali componenti della funzione di loss e delle metriche di valutazione per i modelli YOLOv8 e YOLOv26..... 70
Figura 3.13 Andamento delle componenti della funzione di loss durante il training per i modelli YOLOv8 (a) e YOLOv26 (b)..... 71
Figura 3.14 Valori finali delle principali metriche di valutazione per i modelli YOLOv8 e YOLOv26..... 74
Figura 3.15 Andamento delle metriche di valutazione durante il training: (a) YOLOv8, (b) YOLOv26..... 75
Figura 3.16 Matrice di confusione della rete YOLOv26..... 76
Figura 3.17 Confronto qualitativo tra le prestazioni della reteYOLOv8 (a) e YOLOv26 (b)..... 78
Figura 4.1 Schema del funzionamento del paradigma tracking-by-detection..... 79
Figura 4.2 Schema del funzionamento di BoT-SORT: le detection vengono associate alle tracce tramite una combinazione di predizione del moto (Kalman filter), informazioni geometriche (IoU) e feature di apparenza (ReID)..... 82
Figura 4.3 Pipeline dell'algoritmo ByteTrack: le detection generate dal detector vengono suddivise in alta e bassa confidenza e associate alle tracce in due fasi successive, utilizzando una predizione del moto basata su filtro di Kalman e un processo di assegnazione ottimale..... 84

Figura 5.1 Esempio di tracking multi-oggetto in una sequenza laparoscopica: il sistema rileva e traccia simultaneamente strumenti chirurgici, assegnando identificatori persistenti (ID) e punteggi di confidenza a ciascuna detection. 86
Figura 7.1 Grafici rappresentanti le metriche cinematiche durante la fase di esposizione della colecisti: pinza destra (a) e pinza sinistra (b)..... 95
Figura 7.2 Grafici rappresentanti le metriche cinematiche della fase di isolamento: pinza (a), uncino (b) 98
Figura 7.3 Grafici rappresentanti le metriche cinematiche fase di isolamento: clipper (a), pinza (b), forbici (c)..... 102
Figura 7.4 Grafici rappresentanti le metriche fase di scollamento caso 4 104

1. Introduzione

La chirurgia laparoscopica costituisce oggi il principale riferimento nella chirurgia mininvasiva. Grazie all'impiego di una telecamera e di strumenti inseriti tramite piccole incisioni, è possibile eseguire interventi chirurgici senza ricorrere alla chirurgia open tradizionale. Questo approccio offre numerosi vantaggi per il paziente, tra cui una riduzione del rischio di complicanze, minore dolore post-operatorio, tempi di recupero più rapidi e cicatrici meno evidenti. Nonostante tali benefici, l'approccio laparoscopico presenta alcune limitazioni tecniche. In particolare, l'utilizzo di strumenti rigidi riduce i gradi di libertà di movimento del chirurgo rispetto alla chirurgia open. Inoltre, la visualizzazione del campo operatorio tramite monitor bidimensionale può compromettere la percezione della profondità e delle distanze tra gli strumenti chirurgici e i tessuti circostanti.

Tra gli interventi più comunemente eseguiti con tecnica laparoscopica vi è la colecistectomia laparoscopica, che rappresenta spesso uno dei primi interventi affrontati dai medici in formazione specialistica. In tale contesto, emerge la necessità di disporre di strumenti oggettivi per la valutazione delle prestazioni chirurgiche; il presente elaborato di tesi si propone pertanto di sviluppare un sistema idoneo alla loro analisi.

Attualmente, la valutazione delle competenze chirurgiche si basa prevalentemente su indicatori qualitativi, derivanti dal giudizio soggettivo di chirurghi con ampia esperienza, ovvero aventi una consolidata esperienza nella colecistectomia laparoscopica, intesa come elevata casistica operatoria, documentata autonomia nell'esecuzione della procedura, ridotti tempi operatori, basso tasso di conversione a chirurgia open e limitata incidenza di complicanze intra- e post-operatorie.

Questo approccio presenta diverse limitazioni, tra cui la dipendenza dall'esperienza dell'osservatore, la mancanza di metriche quantitative ~~e~~ la difficoltà nel confrontare in modo oggettivo le prestazioni di operatori diversi.

Alla luce di queste criticità, l'obiettivo del presente lavoro di tesi è sviluppare una pipeline automatizzata per l'analisi quantitativa delle performance chirurgiche a partire dall'analisi di video intraoperatori.

La scelta di basare la valutazione sull'analisi delle registrazioni video risponde all'esigenza di disporre di uno strumento oggettivo che sia al contempo non invasivo e pienamente compatibile con la pratica clinica. In particolare, l'obiettivo è quello di osservare e analizzare l'operato del chirurgo in condizioni quanto più possibile aderenti alla realtà della sala operatoria, evitando l'introduzione di elementi che possano alterare il comportamento degli operatori o il flusso procedurale.

L'impiego di sensori indossabili o integrati negli strumenti chirurgici, pur offrendo dati quantitativi ad alta risoluzione (ad esempio relativi a movimento, forza o traiettorie), presenta diverse limitazioni. Da un lato, tali dispositivi possono risultare invasivi, influenzando la libertà di movimento del chirurgo e introducendo un potenziale bias nelle prestazioni osservate. Dall'altro, pongono problematiche legate alla sterilità, alla complessità di installazione e alla necessità di calibrazione, rendendo più difficoltosa la loro integrazione routinaria in ambiente clinico. Inoltre, l'accettabilità di tali sistemi da parte del personale sanitario può essere limitata, soprattutto in contesti ad alta intensità operativa.

Al contrario, l'analisi video consente di sfruttare una fonte informativa già disponibile, in particolare nella chirurgia laparoscopica, dove la registrazione del campo operatorio rappresenta spesso una pratica standard. Questo approccio permette di acquisire dati senza interferire con l'intervento, preservando l'ecologia della sala operatoria e garantendo una valutazione delle prestazioni in condizioni reali. Un ulteriore vantaggio risiede nella possibilità di effettuare analisi retrospettive, confronti tra operatori e procedure, nonché nell'elevata scalabilità del metodo.

Infine, il video fornisce una rappresentazione diretta e contestualizzata dell'atto chirurgico, permettendo di cogliere non solo aspetti quantitativi, ma anche qualitativi della performance, come la fluidità dei gesti, la gestione delle fasi critiche e l'interazione con il campo operatorio. Tali caratteristiche rendono l'analisi video una scelta particolarmente appropriata per lo sviluppo di sistemi di valutazione oggettiva delle prestazioni chirurgiche.

A tal fine, sono state utilizzate tecniche di computer vision e deep learning per il tracciamento degli strumenti chirurgici durante l'intervento. In particolare, sono state

impiegate tecniche di *deep learning*, progettate per rilevare gli strumenti e seguirne le traiettorie nel tempo. L'addestramento della rete è stato effettuato utilizzando un dataset creato ad hoc tramite un processo di annotazione semi-automatico, consentendo di definire *bounding box* e *label* specifiche per ciascuna tipologia di strumento chirurgico. A partire dai dati ottenuti dal tracciamento degli strumenti, è stata successivamente condotta un'analisi statistica finalizzata all'individuazione di parametri quantitativi associati al raggiungimento di un livello di competenza paragonabile a quello degli operatori esperti.

I risultati ottenuti suggeriscono che le metriche cinematiche bidimensionali sono in grado di caratterizzare il gesto chirurgico e distinguere il livello di esperienza dell'operatore, evidenziando il potenziale utilizzo di tali metodologie come strumenti oggettivi per la valutazione e il supporto alla formazione chirurgica.

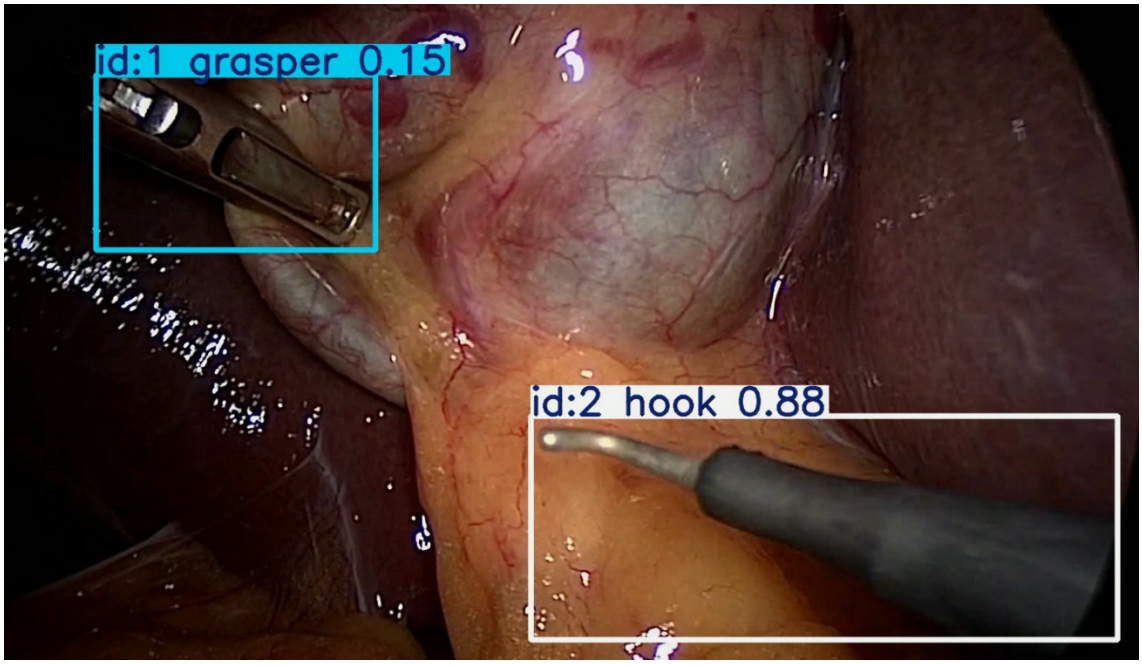


Figura 1.1 Tracking strumenti laparoscopici

1.1. Stato dell'arte

Negli ultimi anni, il problema del *multi-object tracking* (MOT) ha assunto un ruolo centrale nell'ambito della visione artificiale, trovando applicazione anche nel contesto chirurgico per il tracciamento degli strumenti laparoscopici. In generale, il MOT consente di modellare la dinamica temporale degli oggetti, superando i limiti dei metodi basati esclusivamente sul rilevamento statico.

Una panoramica sistematica delle principali tecniche di tracking è fornita da Luo et al. [1], che analizzano nel dettaglio le architetture *tracking-by-detection*. In tali approcci, il processo è tipicamente suddiviso in due fasi: una fase di rilevamento, in cui un modello di *object detection* identifica gli oggetti frame per frame, e una fase di associazione temporale, in cui le *detection* vengono collegate tra frame consecutivi mediante algoritmi di *data association*, come l'assegnamento basato su distanza o su metriche di similarità tra feature visive. Luo et al. evidenziano come la qualità del *tracking* dipenda fortemente sia dall'accuratezza del detector sia dalla robustezza del meccanismo di associazione, in particolare in presenza di occlusioni e oggetti visivamente simili.

Nel dominio specifico della chirurgia laparoscopica, Nwoye et al. propongono SurgiTrack [2], un sistema di *tracking* multi-classe progettato per il tracciamento simultaneo di più strumenti chirurgici. Il metodo si basa su un approccio *tracking-by-detection* in cui le *bounding box* degli strumenti vengono prima rilevate tramite un modello di *detection* e successivamente associate nel tempo per mantenere l'identità degli strumenti. Una caratteristica rilevante del lavoro è l'attenzione al livello fine delle classi e alla gestione di scenari complessi, ottenuta attraverso strategie di associazione robuste e l'utilizzo di feature spaziali e temporali. Tuttavia, gli autori evidenziano come il sistema possa degradare in presenza di occlusioni prolungate o variazioni visive marcate, che compromettono la continuità delle traiettorie.

Parallelamente, la disponibilità di dataset specifici per il dominio chirurgico ha rappresentato un fattore chiave per il progresso del settore. In questo contesto, CholecTrack20 [3] introduce un dataset multi-prospettiva che include sequenze video annotate con *bounding box* e identità persistenti degli strumenti lungo il tempo. Il *dataset* consente di valutare le prestazioni dei modelli di tracking mediante metriche standard del MOT, come MOTA e ID-switches, fornendo un *benchmark* strutturato per il confronto tra

diversi approcci. Nonostante ciò, la variabilità intra-operatoria e la complessità delle condizioni reali rendono ancora difficile ottenere dataset sufficientemente ampi e rappresentativi.

Accanto al problema del tracking, un ruolo fondamentale è svolto dalle tecniche *di object detection*. In particolare, i modelli della famiglia YOLO sono ampiamente utilizzati per il riconoscimento degli strumenti chirurgici grazie alla loro architettura a singolo stadio (single-stage), che consente di effettuare rilevamento e classificazione in tempo reale. La review proposta da Raja et al. [4] analizza le prestazioni delle versioni più recenti di YOLO nel contesto della colecistectomia laparoscopica, evidenziando come tali modelli utilizzino *backbone* convoluzionali profondi e strategie di feature fusion per migliorare la *detection* di oggetti di piccole dimensioni. Tuttavia, le prestazioni risultano fortemente dipendenti dalla qualità e dalla specificità dei dati di addestramento, soprattutto in presenza di rumore visivo e artefatti tipici dell'ambiente chirurgico.

Infine, la letteratura relativa alla valutazione delle competenze chirurgiche sottolinea la necessità di introdurre metriche oggettive e riproducibili. Moorthy et al. [5] evidenziano come l'analisi delle *performance* chirurgiche possa essere supportata da indicatori quantitativi derivati dall'osservazione dei movimenti e delle azioni degli operatori, quali tempi di esecuzione, traiettorie e *pattern* di utilizzo degli strumenti. Tuttavia, tali approcci sono spesso basati su valutazioni manuali o semi-automatiche, evidenziando la necessità di sistemi automatizzati in grado di estrarre queste informazioni direttamente dai dati video.

Nel complesso, i lavori analizzati mostrano come le architetture *tracking-by-detection* e i modelli di *object detection* avanzati rappresentino lo stato dell'arte per il tracciamento degli strumenti chirurgici. Tuttavia, permangono criticità legate alla robustezza in scenari complessi, alla disponibilità di dataset adeguati e alla difficoltà di integrare in modo sistematico il tracking con la valutazione oggettiva delle performance chirurgiche.

1.2. Metodologia proposta

Nel presente lavoro, l'impostazione metodologica si colloca nel paradigma *tracking-by-detection* adottato negli studi più recenti sul tracciamento di strumenti chirurgici, ma se ne discosta per alcune scelte operative. In primo luogo, la decisione di costruire un dataset personalizzato mediante annotazione ad hoc risponde a un'esigenza ben nota in letteratura: nei contesti chirurgici, la qualità e la coerenza delle annotazioni influenzano direttamente la robustezza dei modelli di *detection* e *tracking*, soprattutto in presenza di occlusioni, riflessi, fumo e sangue. I lavori più recenti sul tracking laparoscopico sottolineano infatti come la scarsità di dataset ampi e specificamente annotati costituisca ancora un limite rilevante per l'adozione di sistemi di intelligenza artificiale in questo dominio. In questa prospettiva, la realizzazione di un dataset dedicato ha consentito di adattare le classi e i criteri annotativi alle esigenze del caso di studio, riducendo la dipendenza da schemi di etichettatura troppo generici presenti nei benchmark pubblici [2]. Dal punto di vista operativo, il dataset è stato annotato tramite Roboflow, piattaforma che mette a disposizione strumenti per l'annotazione con *bounding box*, la gestione delle classi e l'organizzazione delle versioni del dataset, aspetti particolarmente utili quando si vuole costruire un impianto coerente e riutilizzabile per compiti di *object detection*. La possibilità di definire e modificare in modo controllato le classi annotate, insieme all'interfaccia dedicata all'annotazione manuale, ha reso questo ambiente adeguato alla costruzione del dataset sperimentale impiegato [6] .

Successivamente è stato scelto l'utilizzo di una rete della famiglia YOLO in ambiente *web-based* coerentemente con l'orientamento della letteratura recente, che evidenzia come tali architetture offrano un compromesso particolarmente favorevole tra accuratezza e velocità di inferenza nelle applicazioni laparoscopiche e, più in generale, nei contesti di *computer vision real-time*. Gli studi sui modelli YOLO applicati al riconoscimento di strumenti in chirurgia laparoscopica mostrano infatti che queste reti risultano particolarmente adatte quando è necessario coniugare buone prestazioni di *detection* con efficienza computazionale. Il modello è stato utilizzato in un ambiente *web-based*, che ha consentito di semplificare le fasi di addestramento, test e validazione del sistema. Questa scelta ha permesso una gestione più immediata delle operazioni sperimentali e una rapida

iterazione tra le diverse configurazioni del modello, mantenendo al contempo un buon livello di controllo sui risultati ottenuti [4,7].

Un aspetto cardine del presente studio è rappresentato dal collegamento tra il tracking degli strumenti e la valutazione delle performance chirurgiche. La letteratura riguardante le abilità tecniche in chirurgia sottolinea la necessità di passare da valutazioni puramente soggettive a metriche oggettive, riproducibili e validate. Su questa base, l'estrazione di metriche bidimensionali derivate dal *tracking* come traiettorie, tempi di permanenza, continuità dei movimenti e pattern di utilizzo degli strumenti è stata orientata non solo alla descrizione tecnica della scena, ma anche alla valutazione della qualità dell'esecuzione chirurgica [5].

Rispetto ai contributi presenti in letteratura, il lavoro proposto si distingue per tre elementi principali: la costruzione di un dataset annotato ad hoc, l'impiego di una rete YOLO in un ambiente *web-based* per la fase di *detection*, e l'utilizzo delle informazioni di *tracking* come base per una valutazione oggettiva delle competenze chirurgiche. Tale impostazione consente di collocare lo studio in continuità con i lavori più recenti sul tracciamento degli strumenti, ma allo stesso tempo di orientarlo verso una finalità applicativa legata all'analisi della performance, fornendo degli strumenti quantitativi che permettano di discriminare le performance di un chirurgo [2].

1.3. Metriche di interesse

L'analisi dei movimenti degli strumenti chirurgici rappresenta un elemento fondamentale nello studio quantitativo delle performance operatorie. In particolare, le traiettorie, la velocità e la precisione dei movimenti costituiscono indicatori significativi per la caratterizzazione del gesto chirurgico e per la distinzione tra diversi livelli di abilità degli operatori. Grazie alla disponibilità dei video laparoscopici, è possibile applicare metodologie di analisi automatizzate per estrarre informazioni cinematiche direttamente dai movimenti degli strumenti durante la procedura, nell'ambito della *surgical data science* [8].

Nel contesto del presente lavoro di tesi, il tracciamento automatico degli strumenti chirurgici nei video operatori consente di ricostruire le traiettorie temporali degli strumenti e di calcolare specifiche metriche cinematiche utili alla valutazione delle prestazioni. Dal punto di vista matematico, tali metriche possono essere derivate a partire dalla traiettoria dello strumento nel tempo x(t), da cui è possibile ottenere velocità, accelerazione e jerk come derivate successive rispetto al tempo. L'utilizzo di queste grandezze è ampiamente documentato in letteratura per la valutazione oggettiva delle competenze chirurgiche [9,10].

Le principali metriche considerate nel presente studio sono le seguenti:

- Velocità dei movimenti: definita come la derivata prima della posizione rispetto al tempo, rappresenta la rapidità con cui lo strumento si sposta nello spazio. In letteratura, velocità eccessivamente elevate o non uniformi sono spesso associate a un minore livello di esperienza, mentre operatori esperti tendono a mantenere una velocità più controllata e coerente con il contesto operativo[9].
- Accelerazione: derivata seconda della traiettoria, descrive le variazioni della velocità nel tempo. Valori elevati possono indicare correzioni improvvise della traiettoria o difficoltà nell'interazione con i tessuti, mentre andamenti più regolari sono generalmente associati a movimenti più stabili e controllati [10].
- Jerk: derivata terza della traiettoria, rappresenta una misura della fluidità del movimento. Valori ridotti di jerk indicano movimenti continui e privi di brusche

variazioni, tipicamente associati a un maggiore livello di esperienza chirurgica, mentre valori elevati riflettono movimenti discontinui o poco fluidi [9].

- Distanza totale percorsa: misura la lunghezza complessiva della traiettoria dello strumento. Percorsi più diretti ed efficienti sono generalmente associati a operatori esperti, mentre percorsi più lunghi e tortuosi possono indicare esitazioni o correzioni frequenti [10].
- Tempo di esecuzione dei gesti: rappresenta la durata necessaria per completare specifiche fasi operative. Sebbene tempi ridotti possano indicare efficienza, è necessario considerarli congiuntamente ad altre metriche, poiché tempi leggermente superiori, se associati a movimenti fluidi e controllati, possono riflettere una maggiore accuratezza e consapevolezza operativa [8].

L'analisi congiunta di queste metriche è stata condotta seguendo un approccio standardizzato per ciascuna fase dell'intervento, al fine di consentire un confronto sistematico tra i movimenti eseguiti da chirurghi esperti e da medici in formazione. In particolare, durante l'isolamento della colecisti è fondamentale che i movimenti risultino precisi e controllati, data la prossimità a strutture anatomiche critiche. Nella fase di legatura e sezionamento del dotto cistico e dell'arteria cistica, la stabilità degli strumenti, la corretta angolazione e la fluidità dei movimenti rappresentano elementi determinanti per la sicurezza della procedura. Infine, durante la rimozione della colecisti, assumono particolare rilevanza la coordinazione occhio-mano e la regolarità delle traiettorie.

Nel complesso, l'analisi delle metriche cinematiche nelle diverse fasi dell'intervento consente di individuare pattern distintivi del comportamento operatorio, rendendo possibile una valutazione oggettiva e quantitativa delle competenze chirurgiche. Questo approccio permette di superare i limiti delle valutazioni soggettive tradizionali, contribuendo allo sviluppo di sistemi automatici di supporto alla formazione e alla valutazione clinica.

1.4. Colecistectomia laparoscopica

Il seguente capitolo offre una panoramica anatomica e funzionale della colecisti, descrivendo le principali fasi relative alla sua asportazione. La trattazione è volta a contestualizzare le metriche cinematiche di ogni processo, mettendo in relazione i movimenti degli strumenti con le specifiche manovre chirurgiche eseguite durante l'intervento. Infatti, l'approfondimento dell'anatomia della colecisti e delle strutture adiacenti risulta essere fondamentale per la comprensione della successiva analisi automatica delle procedure e della valutazione delle performance chirurgiche.

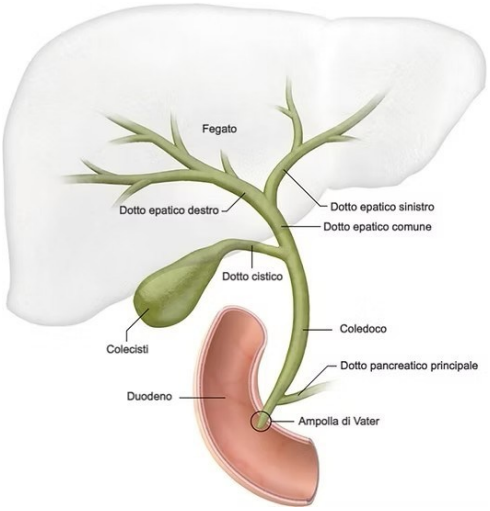


Figura 1.2 Anatomia della colecisti

1.4.1. Anatomia della colecisti

La colecisti è un organo cavo facente parte del sistema biliare, localizzato sulla superficie inferiore del fegato, all'interno della fossa cistica. Dal punto di vista anatomico, essa può essere suddivisa in tre porzioni principali: fondo, corpo e collo. Il fondo rappresenta la parte più distale dell'organo e in alcuni casi può protrudere leggermente oltre il margine inferiore del fegato, il corpo costituisce la porzione centrale a contatto con la superficie viscerale epatica; infine, il collo prosegue nel dotto cistico, mettendo in comunicazione la colecisti con le vie biliari extraepatiche

[11]. Il dotto cistico si unisce al dotto epatico comune formando il coledoco, che convoglia la bile verso il duodeno. In prossimità del collo della colecisti decorre generalmente l'arteria cistica, che nella maggior parte dei casi origina dall'arteria epatica destra e rappresenta la principale fonte di vascolarizzazione della colecisti.

Un punto anatomico particolarmente rilevante durante l'intervento chirurgico è il cosiddetto triangolo di Calot (Fig.1.3), delimitato dal dotto cistico, dal dotto epatico comune e dal margine inferiore del fegato. All'interno di questa regione decorre l'arteria cistica, insieme a piccoli rami vascolari e strutture linfatiche. La corretta identificazione di tali strutture anatomiche è fondamentale durante la dissezione chirurgica nel corso della colecistectomia laparoscopica, al fine di evitare lesioni delle vie biliari principali.

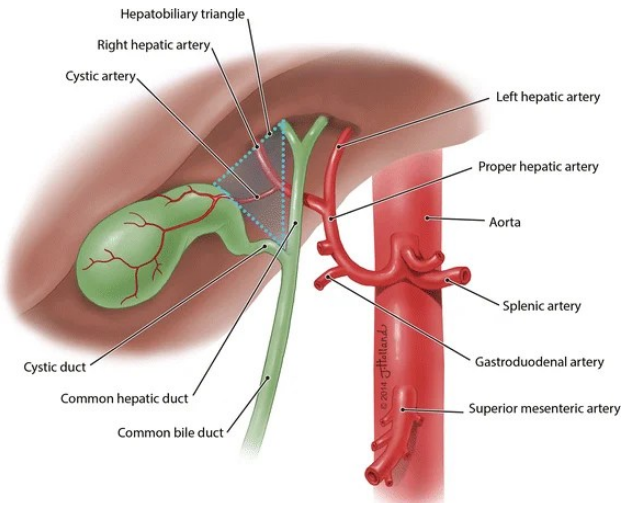


Figura 1.3 Triangolo di Calot

Dal punto di vista funzionale, la colecisti svolge un ruolo essenziale nei processi digestivi, in particolare nella digestione e nell'assorbimento dei lipidi. La bile, prodotta continuamente dagli epatociti del fegato, viene convogliata attraverso i dotti epatici nelle vie biliari. In condizioni di digiuno, una parte significativa della bile viene deviata attraverso il dotto cistico e immagazzinata nella colecisti, dove subisce un processo di concentrazione dovuto all'assorbimento di acqua ed elettroliti da parte dell'epitelio della mucosa colecistica. Questo processo consente di aumentare la concentrazione dei sali biliari e degli altri componenti della bile, rendendola più efficace nei processi digestivi

[12]. Durante l'assunzione di cibo, in particolare in presenza di lipidi nel lume intestinale, le cellule endocrine della mucosa duodenale rilasciano colecistochinina, un ormone che stimola la contrazione della parete muscolare della colecisti e il contemporaneo rilascio dello sfintere di Oddi. Questo meccanismo coordinato determina lo svuotamento della colecisti e il passaggio della bile nel duodeno attraverso il coledoco. La bile contribuisce all'emulsione dei grassi alimentari, facilitando l'azione delle lipasi pancreatiche e favorendo l'assorbimento intestinale dei lipidi e delle vitamine liposolubili [12].

Dal punto di vista clinico, la rimozione della colecisti rappresenta uno degli interventi chirurgici maggiormente eseguiti. Tale procedura viene spesso eseguita in presenza di colelitiasi, condizione caratterizzata dalla formazione di calcoli all'interno della colecisti. I calcoli biliari si formano generalmente a causa di alterazioni nella composizione della bile, in particolare per un aumento della concentrazione di colesterolo o per squilibri nei sali biliari e nei fosfolipidi, che portano alla precipitazione di cristalli e alla successiva aggregazione in concrezioni solide [12]. Nella maggior parte dei casi la colelitiasi può essere asintomatica; tuttavia, quando i calcoli ostruiscono temporaneamente il dotto cistico, possono provocare episodi di colica biliare, caratterizzati da dolore intenso localizzato nel quadrante superiore destro dell'addome. In alcuni casi, l'ostruzione persistente del dotto cistico può determinare l'insorgenza di colecistite acuta, una condizione infiammatoria della colecisti che rappresenta una delle indicazioni più comuni alla colecistectomia [12]. Ulteriori complicanze associate alla presenza di calcoli biliari includono la migrazione dei calcoli nelle vie biliari principali, con conseguente coledocolitiasi, che può causare ostruzione del coledoco e determinare ittero ostruttivo. In casi più gravi, l'ostruzione delle vie biliari può favorire lo sviluppo di infezioni delle vie biliari, come la colangite, oppure contribuire all'insorgenza di pancreatite acuta biliare, dovuta all'ostruzione transitoria della regione ampollare a livello del duodeno [12].

Alla luce di queste possibili complicanze, la colecistectomia rappresenta il trattamento di riferimento nei pazienti sintomatici o nei casi in cui vi sia un elevato rischio di evoluzione clinica sfavorevole. La rimozione della colecisti consente infatti di eliminare la sede di formazione dei calcoli e prevenire il ripetersi degli episodi dolorosi e delle complicanze associate. Oggi, nella maggior parte dei casi, tale procedura viene eseguita mediante

approccio laparoscopico, che consente di ridurre il trauma chirurgico e favorire un recupero post-operatorio più rapido rispetto alla chirurgia open

1.4.2. Procedura di rimozione della colecisti

L'intervento di colecistectomia consta di una serie di passaggi sequenziali che comprendono l'accesso alla cavità addominale, l'identificazione e la dissezione delle strutture anatomiche chiave, la legatura e la sezione del dotto e dell'arteria cistica, lo scollamento della colecisti dal letto epatico e la sua estrazione, con l'obiettivo di asportare la colecisti in sicurezza minimizzando il rischio di complicanze post-operatorie.

L'accesso laparoscopico prevede tipicamente l'introduzione di 4 trocar nella parete addominale. In particolare, secondo la tecnica americana:

1. Trocar ombelicale (12 mm) per l'inserimento della telecamera, attraverso la quale si ottiene la visione del campo operatorio.
2. Trocar epigastrico (5 mm) per il passaggio degli strumenti principali della mano destra, utilizzato per la dissezione e la coagulazione.
3. Trocar sottocostale destro (5 mm) per ulteriori strumenti accessori della mano sinistra, secondo necessità del chirurgo.
4. Trocar in fianco destro (5 mm) per strumenti ausiliari, che facilitano il sollevamento del fegato e la trazione della colecisti.

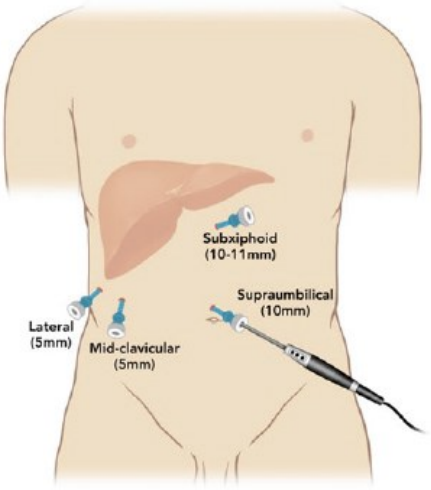


Figura 1.4 Posizionamento dei trocar per l'intervento di colecistectomia laparoscopica

La disposizione dei trocar e degli strumenti determina la configurazione del campo visivo laparoscopico, influenzando direttamente la posizione, l'orientamento e la visibilità degli strumenti chirurgici. Tutti questi aspetti risulteranno rilevanti nella fase di analisi, in quanto condizionano la qualità del rilevamento e del tracciamento automatico degli strumenti.

Per l'inserimento dei trocar, la cavità addominale viene insufflata mediante ago di Verres con anidride carbonica (CO_2) per creare uno spazio operativo sufficiente alla manipolazione degli organi e degli strumenti. L'accesso laparoscopico consente una visione ingrandita e illuminata del campo chirurgico mediante un monitor ad alta risoluzione. La limitazione che si riscontra andando a visualizzare gli organi mediante un video è data da una rappresentazione bidimensionale della scena, con conseguente perdita di informazioni relative alla profondità. Questa caratteristica rappresenta una delle principali criticità sia per il chirurgo sia per i sistemi automatici di analisi video, che devono risalire a relazioni spaziali tra strumenti e tessuti a partire da immagini 2D.

Una volta identificata la colecisti, si procede all'isolamento delle strutture presenti in prossimità del collo. In questa fase, il fondo della colecisti viene trazionato verso l'alto, in modo da permettere l'esposizione del legamento epato-duodenale (composto da due foglietti peritoneali del piccolo omento) ed evidenziare in trasparenza il coledoco e l'arteria epatica destra.

Successivamente, il dotto cistico e l'arteria cistica vengono progressivamente isolati dal legamento epato-duodenale mediante l'uso di un elettrodo a uncino. La mobilizzazione delle strutture è completata utilizzando un dissettore curvo, consentendo di liberare delicatamente i tessuti circostanti. In questa fase, il movimento coordinato degli strumenti e la presenza di occlusioni parziali tra strumenti e tessuti rendono particolarmente complesso il tracciamento poiché si deve mantenere l'identità degli strumenti anche in presenza di sovrapposizioni e cambiamenti rapidi di posizione [13].

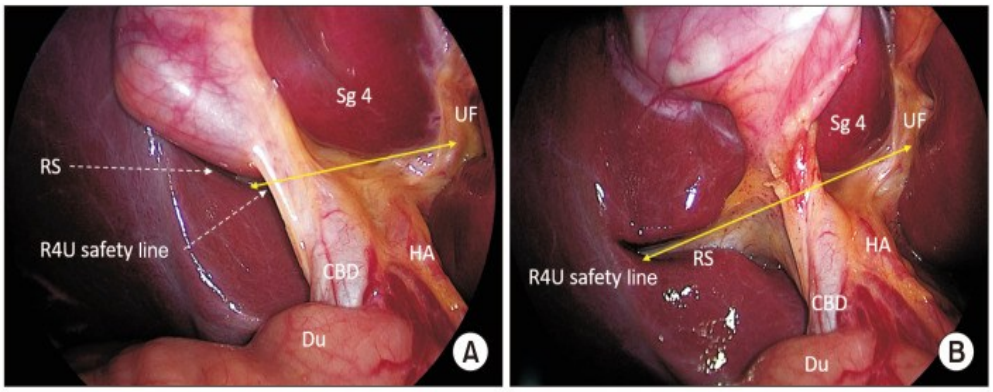


Figura 1.5 Campo chirurgico in vista anteriore/mediale (A) e vista posteriore/laterale (B) del triangolo epatocistico e delle aree adiacenti. CBD: dotto biliare comune; Du. duodeno; AH: arteria epatica; RS: solco di Rouvière; Sg4: segmento 4; UF: fessura ombelicale

L'isolamento delle strutture biliari avviene tramite dissezione progressiva del triangolo di Calot, che permette di identificare chiaramente il dotto cistico e l'arteria cistica, strutture che dovranno essere legate e sezionate nelle fasi successive. Durante questa manovra è fondamentale procedere con estrema attenzione, evitando trazioni eccessive o movimenti bruschi degli strumenti, poiché una visualizzazione incompleta delle strutture anatomiche può aumentare il rischio di lesioni delle vie biliari principali, una delle complicanze più gravi associate alla colecistectomia laparoscopica. Per minimizzare tale rischio, nella pratica chirurgica si applica il principio della *Critical View of Safety*, ampiamente descritto in letteratura come standard di riferimento per la colecistectomia laparoscopica, che richiede di ottenere una chiara visualizzazione del dotto cistico e dell'arteria cistica, assicurandosi che solo queste due strutture siano collegate alla colecisti prima di procedere alla loro sezione. Dal punto di vista della visione artificiale, questa fase rappresenta un momento chiave, in cui la corretta identificazione delle strutture anatomiche e delle interazioni con gli strumenti risulta fondamentale per l'analisi automatica della procedura.

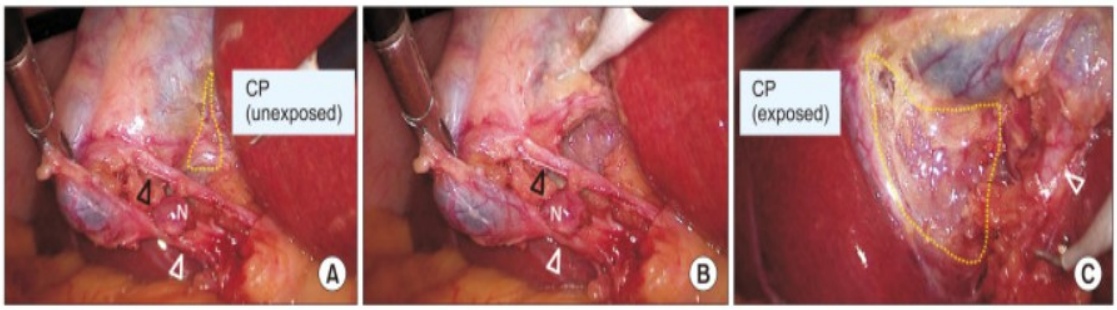


Figura 1.6 (A) Il triangolo di Calot viene dissezionato per esporre e delimitare il dotto cistico e l'arteria cistica. Il piano cistico non è ancora esposto (linea rotta). (B)Il piano cistico viene esposto. (C) La CP è esposta adeguatamente (linea interrotta). La punta bianca indica il dotto cistico e la punta nera indica l'arteria cistica. PC, placca cistica

Dopo aver isolato correttamente il dotto cistico e l'arteria cistica nel triangolo di Calot, il passo successivo consiste nella loro legatura e sezionamento. Il chirurgo applica generalmente due clip chirurgiche sul dotto cistico e sull'arteria cistica, posizionando con attenzione per evitare qualsiasi coinvolgimento accidentale di tessuti circostanti. Successivamente si procede con la sezione degli stessi condotti mediante forbici laparoscopiche. Le traiettorie degli strumenti in questa fase risultano particolarmente informative, in quanto caratterizzate da movimenti precisi e ripetitivi. Tali pattern possono essere utilizzati per l'estrazione di metriche cinematiche utili alla valutazione delle performance chirurgiche.

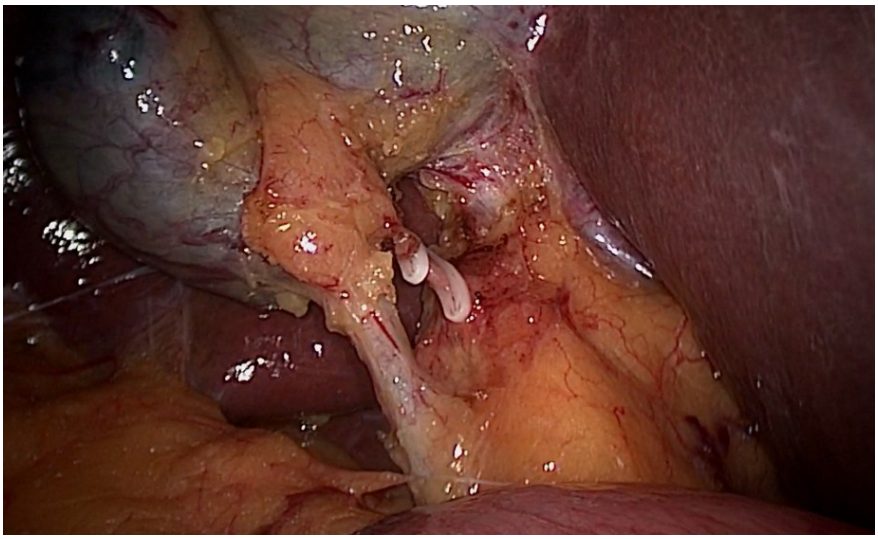


Figura 1.7 Applicazione delle clip sul dotto cistico e sull'arteria cistica durante colecistectomia laparoscopica

Nell'ultima fase la colecisti viene staccata dal parenchima epatico utilizzando strumenti laparoscopici, generalmente con movimenti di scollamento lenti e controllati lungo il piano di aderenza. Una volta completamente liberata, la colecisti viene posizionata in un sacchetto sterile e rimossa attraverso uno dei trocar. Questo passaggio richiede particolare cautela: se la colecisti si perfora o fuoriesce della bile o dei calcoli, possono insorgere complicanze come infezioni o aderenze.

Nel complesso, la sequenza degli step chirurgici descritti fornisce il contesto operativo entro il quale si inserisce il presente lavoro, in cui il tracciamento automatico degli strumenti viene utilizzato per analizzare quantitativamente il gesto chirurgico e supportare la valutazione delle competenze degli operatori.

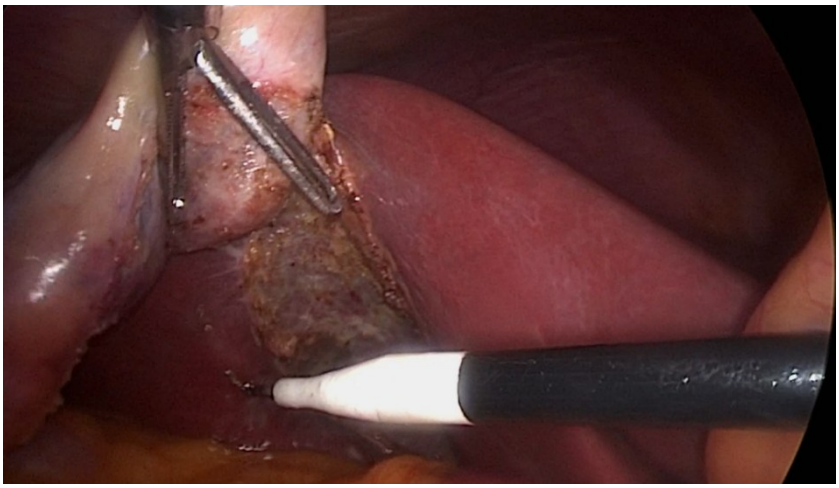


Figura 1.8 Dissezione della colecisti dal letto epatico durante colecistectomia laparoscopica.

2. Annotazione degli strumenti chirurgici mediante Roboflow

L'integrazione di tecniche di *Computer Vision* nei contesti chirurgici rappresenta uno degli sviluppi più significativi dell'ingegneria biomedica applicata alla sala operatoria. In particolare, la chirurgia laparoscopica, caratterizzata da un accesso mininvasivo e dalla visualizzazione indiretta del campo operatorio tramite telecamera endoscopica, genera grandi quantità di dati video che possono essere analizzati mediante algoritmi di deep learning per supportare attività come il riconoscimento automatico degli strumenti, l'identificazione delle traiettorie e la valutazione delle performance operatorie [8]. In tale scenario, la disponibilità di dataset accuratamente annotati costituisce un prerequisito fondamentale per l'addestramento e la validazione di modelli di *object detection* robusti e affidabili [9,10].

Le immagini laparoscopiche presentano caratteristiche peculiari che rendono complessa l'applicazione di tecniche automatiche di rilevamento degli oggetti. Tra queste si sottolineano l'illuminazione non uniforme, la presenza di riflessi speculari sugli strumenti metallici, le frequenti occlusioni parziali, il fumo chirurgico e la variabilità morfologica degli strumenti appartenenti alla medesima categoria funzionale. Dal punto di vista tecnologico, i moderni approcci di *object detection* basati su reti neurali convoluzionali richiedono annotazioni spaziali precise, generalmente sotto forma di *bounding box* con *label* associata, per apprendere la localizzazione e la classificazione degli oggetti di interesse [14–16].

La fase di etichettatura rappresenta uno snodo centrale dell'intero workflow sperimentale. Essa si colloca a valle della raccolta e selezione dei frame laparoscopici e a monte della fase di addestramento del modello di deep learning. Il processo può essere sintetizzato nelle seguenti macro-fasi:

- acquisizione e preprocessing preliminare delle immagini
- annotazione manuale mediante bounding box e definizione delle classi
- validazione e revisione delle etichette
- esportazione del dataset nel formato compatibile con il framework di addestramento scelto.

L’obiettivo del seguente lavoro di tesi è il riconoscimento e la localizzazione di strumenti laparoscopici all’interno di frame estratti da video chirurgici di colecistectomia (1). La qualità delle annotazioni assume pertanto un ruolo centrale, in quanto errori nella definizione delle etichette o nell’estensione delle *bounding box* possono tradursi in una riduzione significativa delle prestazioni del modello in termini di *Intersection over Union (IoU)* e *mean Average Precision (mAP)*.

Il seguente capitolo descrive in modo sistematico il processo di costruzione del dataset mediante l’impiego della piattaforma Roboflow per la creazione delle etichette semantiche e per la creazione delle *bounding box* sugli strumenti laparoscopici. La trattazione è volta a fornire una descrizione metodologicamente rigorosa delle scelte adottate durante la fase di annotazione (definizione delle classi, criteri di disegno delle *bounding box,* gestione delle ambiguità visive) ed evidenziare le implicazioni di tali scelte sulla qualità del dataset e, conseguentemente, sulle prestazioni del modello di rilevamento addestrato nelle fasi successive della tesi.

2.1. La piattaforma Roboflow

La gestione di dataset per *object detection* richiede strumenti in grado di garantire accuratezza annotativa, tracciabilità delle versioni e compatibilità con i principali framework di *deep learning*. In questo contesto, Roboflow rappresenta una piattaforma integrata progettata per facilitare la creazione, la gestione e l'esportazione di dataset di immagini annotati, con particolare attenzione alle applicazioni di visione artificiale e *machine learning*.

Roboflow offre un ambiente *web-based* che consente di importare immagini da diverse fonti, definire classi semantiche e annotare manualmente o semi-automaticamente gli oggetti di interesse tramite *bounding box* o segmentazioni. La piattaforma supporta inoltre il *preprocessing* delle immagini, la gestione delle versioni del dataset e l'esportazione nei principali formati compatibili con framework come PyTorch, TensorFlow o YOLO. Tra i vantaggi principali vi è la possibilità di integrare strumenti di assistenza basati su *machine learning* per suggerire automaticamente *bounding box* preliminari, riducendo il contributo manuale e migliorando la coerenza annotativa su grandi dataset [17].

L'architettura di Roboflow può essere concettualmente suddivisa in tre componenti principali:

- Annotazione: consente di disegnare *bounding box*, etichettare classi e gestire strumenti multipli per immagine. La piattaforma offre funzionalità di revisione interna e supporta collaborazioni tra più annotatori, permettendo di consolidare e verificare la qualità delle etichette.
- *Preprocessing*: include operazioni standard di ridimensionamento, normalizzazione, bilanciamento dei colori e applicazione di tecniche di *data augmentation* come rotazioni, riflessioni e modifiche di luminosità. Queste trasformazioni sono essenziali per aumentare la variabilità del dataset, migliorando la capacità del modello di generalizzare su dati nuovi [14,18].
- *Versioning*: Roboflow consente la creazione di versioni del dataset annotate in momenti differenti, consentendo di tracciare modifiche, correggere errori e confrontare diverse configurazioni di *preprocessing*. Questa funzionalità è cruciale per garantire la riproducibilità sperimentale e per documentare le scelte

metodologiche adottate nella costruzione del dataset, particolarmente rilevante in contesti accademici e di ricerca.

Nel complesso, la piattaforma Roboflow fornisce un ecosistema strutturato che semplifica la gestione di dataset complessi, come quelli costituiti da immagini laparoscopiche, assicurando al contempo qualità, tracciabilità e compatibilità con strumenti di addestramento di modelli di *deep learning*.

Figura 2.1 Interfaccia della piattaforma Roboflow

2.2. Creazione del dataset

La costruzione di un dataset accurato costituisce una fase cruciale nel *workflow* di sviluppo di modelli di *object detection*, in quanto la qualità dei dati annotati influenza direttamente le prestazioni del modello [10,19]. Nel presente lavoro, il dataset è costituito a partire da immagini estratte da video di interventi di rimozione della colecisti reali. L'obiettivo è quello di identificare e localizzare con precisione gli strumenti chirurgici presenti in ciascun frame, creando un dataset robusto, coerente e utilizzabile per addestrare modelli di deep learning.

2.2.1. Origine delle immagini

Le immagini del dataset provengono da registrazioni laparoscopiche ad alta risoluzione, acquisite con telecamere endoscopiche standard utilizzate in sala operatoria. Ciascun video è caratterizzato da peculiarità tipiche del contesto clinico, quali illuminazione variabile, riflessi metallici, tessuti sovrapposti e possibili occlusioni parziali degli strumenti. La diversità dei contesti, delle angolazioni e della posizione degli strumenti all'interno del campo visivo rappresenta una sfida per la creazione di annotazioni accurate, ma è fondamentale per garantire la generalizzabilità del modello.

Per costruire il dataset di immagini a partire dai video laparoscopici originali, è stato sviluppato uno script dedicato in ambiente MATLAB. Lo script consente di automatizzare l'estrazione di singoli frame dai video, generando immagini grezze utilizzabili per la successiva fase di selezione e annotazione.

Il procedimento adottato prevede la lettura sequenziale dei frame di ogni video. Ogni frame viene salvato come immagine separata in una cartella di output dedicata, con un sistema di denominazione progressiva standardizzato, garantendo la tracciabilità della sequenza temporale originale. L'automazione di questa fase consente di estrarre tutti i frame disponibili senza intervento manuale, riducendo errori e tempi di elaborazione.

Inoltre, lo script è parametrizzabile: è possibile selezionare il video sorgente, specificare la cartella di destinazione e scegliere il formato di output delle immagini.

Successivamente è stato creato un secondo script in ambiente MATLAB per automatizzare la selezione delle immagini. Il codice è stato progettato per operare su più cartelle sorgenti, ciascuna contenente frame provenienti dai diversi video. Per ogni cartella, lo script effettua:

- individuazione automatica delle immagini compatibili
- verifica della presenza di un numero minimo di immagini;
- selezione casuale uniforme di un numero prefissato di frame;
- copia dei file selezionati in una directory di output comune;
- rinominazione progressiva dei file secondo una convenzione standardizzata.

L'estrazione dei frame dai video laparoscopici è stata effettuata mediante un campionamento casuale controllato, con l'obiettivo di ridurre la ridondanza temporale tipica dei video chirurgici, evitando la presenza eccessiva di frame consecutivi altamente correlati, che potrebbero introdurre *bias* nel training del modello. La selezione dei frame è però avvenuta seguendo criteri specifici, volti a garantire che le immagini incluse fossero di alta qualità e rappresentative delle diverse fasi operatorie. In particolare, sono stati considerati fattori quali la visibilità e la completezza degli strumenti chirurgici, l'illuminazione e la nitidezza dell'immagine, nonché la varietà delle situazioni operative presenti nei video originali; sono state inoltre incluse immagini di solo sfondo, per permettere alla rete di apprendere sia una condizione di presenza che di assenza di strumenti. Inoltre, per assicurare un bilanciamento efficace tra le classi di strumenti, la selezione casuale è stata stratificata in modo da garantire un numero sufficiente di esempi per ciascuna classe, evitando che strumenti più frequenti dominassero il dataset e riducendo il rischio di introdurre *bias* durante l'addestramento del modello. Questo approccio ha permesso di ottenere un dataset non solo statisticamente robusto, ma anche clinicamente rappresentativo, capace di riflettere la reale distribuzione e diversità degli strumenti laparoscopici utilizzati nelle procedure considerate. L'adozione di questa strategia combinata ha consentito di ottimizzare la robustezza delle annotazioni, concentrandosi sulle immagini più informative, e di mantenere la piena riproducibilità del processo.

Il contesto anatomico e la complessità delle strutture coinvolte nella regione epatobiliare contribuiscono alla variabilità delle immagini laparoscopiche

2.2.2. Definizione delle classi

La definizione delle etichette è stata guidata da un'analisi preliminare degli strumenti effettivamente utilizzati nelle procedure considerate. Ai fini dell'addestramento di un modello di *object detection*, è stata privilegiata una classificazione basata sulla riconoscibilità visiva e sulla coerenza morfologica all'interno di ciascuna classe. Tale scelta consente di ridurre l'ambiguità annotativa e di favorire l'apprendimento di caratteristiche visive stabili da parte della rete neurale. Le classi di strumenti laparoscopici definite sono state:

- *clipper (0)*
- *grasper (1)*
- *hook (2)*
- *irrigator (3)*
- *scissor (4)*

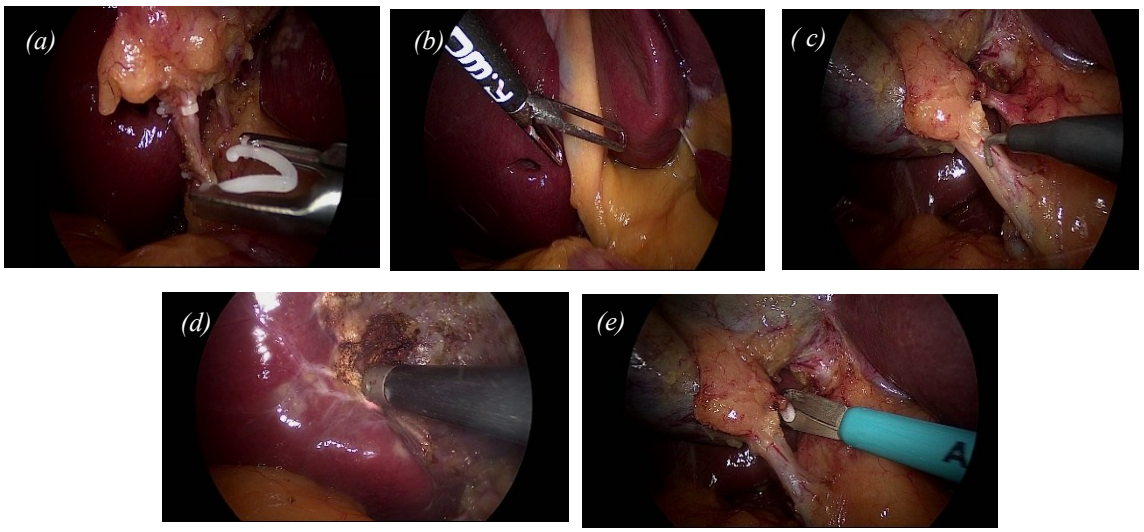


Figura 2.2 Classi strumenti laparoscopici: (a) clipper, (b) grasper, (c) hook, (d) irrigator, (e) scissor

Durante l'annotazione si sono riscontrate alcune criticità. In particolare, le immagini laparoscopiche presentano frequentemente strumenti con caratteristiche visive molto simili, come differenti tipologie di pinze o strumenti con estremità affini ma con funzioni diverse; questa somiglianza può generare ambiguità sia per l'annotatore sia per il modello. Ci sono inoltre situazioni in cui lo strumento è parzialmente visibile, fuori fuoco o coperto da tessuti biologici, rendendo incerta l'identificazione della classe; per garantire coerenza e riproducibilità, sono stati definiti criteri decisionali uniformi: qualora l'identificazione non fosse supportata da evidenze visive sufficienti, l'oggetto non viene annotato oppure classificato nella categoria più generale coerente con le informazioni disponibili. Questo approccio, pur comportando una possibile riduzione del numero totale di annotazioni, contribuisce a migliorare l'affidabilità complessiva del dataset e a ridurre l'introduzione di etichette errate, che avrebbero potuto influenzare negativamente le prestazioni del modello in fase di addestramento.

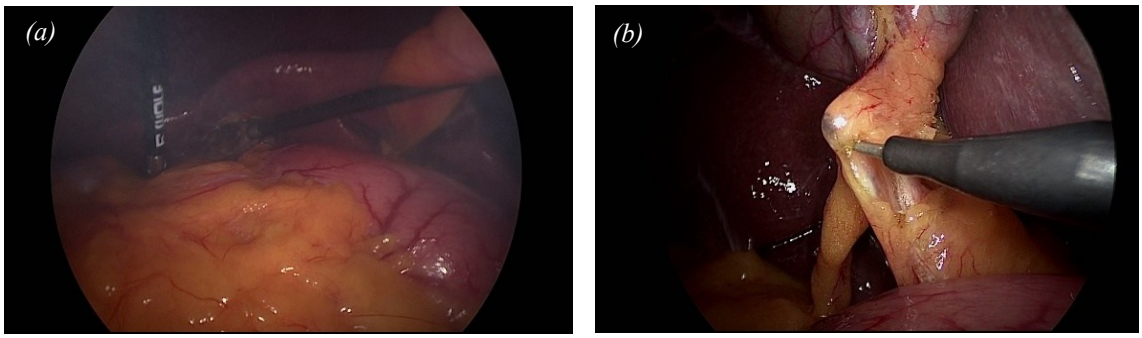


Figura 2.3 Difficoltà di visualizzazione dello strumento chirurgico: (a) frame con scarsa luminosità e obiettivo sporco, (b) frame con strumento parzialmente occluso da tessuto grasso

2.2.3. Creazione delle label e delle bounding box

La fase di etichettatura rappresenta il nucleo metodologico dell'intero processo di costruzione del dataset, in quanto traduce le immagini grezze in dati strutturati utilizzabili per l'addestramento di modelli di *object detection*. Nel presente studio, l'annotazione è stata effettuata mediante *bounding box* rettangolari attraverso la piattaforma Roboflow, seguendo criteri espliciti volti a garantire coerenza, precisione e riproducibilità [10,19–21].

La fase iniziale ha previsto la creazione di un nuovo progetto di tipo *Object Detection*, selezionando il formato di annotazione basato su *bounding box rettangolari*. Successivamente, sono state definite le classi di interesse e caricate le 2000 immagini precedentemente estratte. La piattaforma consente di organizzare le immagini in un unico workspace centralizzato, facilitando la gestione progressiva dell'annotazione e la verifica dello stato di completamento del dataset. L'interfaccia *web-based* ha permesso un accesso strutturato ai frame e una navigazione efficiente tra le immagini da annotare. Le label sono state definite in fase preliminare, assegnando ad ogni classe una specifica colorazione e numerazione, successivamente sono state associate manualmente a ciascuna bounding box durante l'annotazione.

Particolare attenzione è stata dedicata alla definizione dei criteri per il disegno delle *bounding box*, al fine di ridurre la variabilità annotativa. I criteri di disegno sono stati:

- La *bounding box* deve racchiudere interamente la porzione visibile dello strumento.
- I margini devono essere il più possibile aderenti ai contorni dell'oggetto, evitando la presenza di un eccessivo sfondo.
- Non devono essere incluse aree non pertinenti (tessuti, altri strumenti, artefatti visivi).
- In caso di strumenti parzialmente visibili, la *bounding box* delimita esclusivamente la parte osservabile.

Questo approccio mira a fornire al modello informazioni spaziali precise, migliorando l'apprendimento delle caratteristiche morfologiche distintive degli strumenti laparoscopici.

Le immagini laparoscopiche presentano frequentemente condizioni che rendono l'annotazione difficoltosa. Sono stati pertanto definiti criteri specifici per la gestione di:

- Occlusioni parziali: lo strumento viene annotato se la porzione visibile è sufficiente a identificarne la classe con certezza.
- Strumenti multipli nello stesso frame: ciascun strumento viene annotato con una *bounding box* distinta, anche in presenza di sovrapposizione parziale.
- Strumenti fuori fuoco o con *motion blur*[1]: l'annotazione viene effettuata solo se i contorni risultano comunque distinguibili.
- Riflessi e artefatti: i riflessi metallici non vengono considerati oggetti separati e non devono influenzare l'estensione della *bounding box*.

L'adozione di tali criteri ha consentito di mantenere la coerenza lungo l'intero dataset, riducendo il rumore introdotto da decisioni annotative arbitrarie.

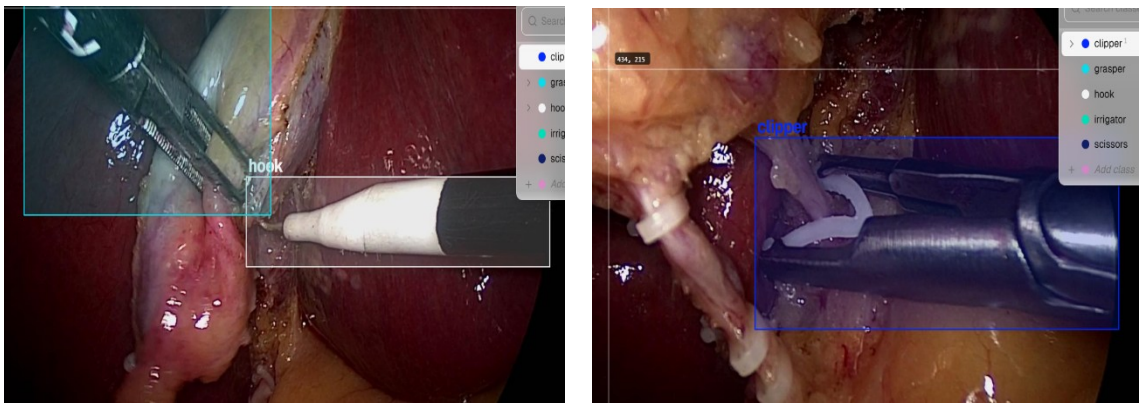


Figura 2.4 Creazione label e bounding box

A seguito della creazione della *bounding box* e dell'assegnazione dell'etichetta per ciascun frame, la piattaforma Roboflow genera automaticamente un file di annotazione in formato *.txt* associato ad ogni immagine del dataset. In particolare, per ciascun frame annotato viene prodotto un file di testo distinto, avente lo stesso nome dell'immagine

[1] Effetto visivo di sfocatura che si genera in caso di movimento di un oggetto mentre viene registrato

corrispondente, contenente le informazioni relative agli oggetti identificati all'interno di quel frame.

2.2.4. Data augmentation

Una volta completate le fasi di annotazione, il dataset è stato sottoposto a operazioni di *data augmentation* tramite la piattaforma Roboflow. Questa fase riveste un ruolo cruciale nel *workflow* di *computer vision* in quanto migliora la capacità di generalizzazione del modello e riduce il rischio di *overfitting*[2]. In ambito laparoscopico, tali tecniche devono essere selezionate con cautela per evitare trasformazioni non realistiche rispetto al contesto clinico. Le operazioni potenzialmente applicabili includono:

- rotazioni limitate (coerenti con l'orientamento reale della telecamera);
- variazioni di luminosità e contrasto;
- flip orizzontale, qualora compatibile con l'interpretazione semantica delle immagini.

L'obiettivo *dell'augmentation* è simulare variazioni plausibili delle condizioni intraoperatorie, aumentando la variabilità del dataset senza alterarne la coerenza clinica. Partendo dalle 2000 immagini annotate inizialmente, è stato implementato un incremento artificiale mediante rotazioni controllate. In particolare, ogni immagine è stata inclinata di 10° in entrambe le direzioni e sono state effettuate ulteriori trasformazioni automatiche offerte dalla piattaforma, generando nuove istanze sintetiche pur mantenendo la coerenza visiva e la correttezza delle annotazioni delle *bounding box*. Questo procedimento ha consentito di espandere il dataset fino a quasi 14.000 immagini totali e 14.860 box, aumentando significativamente la variabilità dei dati a disposizione per l'addestramento del modello senza richiedere ulteriori annotazioni manuali. La scelta di rotazioni limitate è stata motivata dal contesto chirurgico: variazioni angolari eccessive potrebbero generare

[2] condizioni per cui un modello statistico si adatta eccessivamente ai dati di addestramento, apprendendo sia la struttura che il rumore presenti nei dati, con conseguente bassa capacità di generalizzazione.

configurazioni non realistiche nei video laparoscopici, compromettendo la fedeltà clinica del dataset. L'utilizzo della *data augmentation* ha quindi permesso di:

- aumentare il numero di esempi disponibili per ciascuna classe di strumento
- ridurre il rischio di *overfitting*, soprattutto per le classi meno rappresentate
- Mantenere la coerenza con le condizioni visive realistiche dei video originali.

Tale strategia rappresenta l'unico intervento di incremento artificiale applicato al dataset e costituisce un elemento chiave per garantire robustezza e generalizzazione del modello nelle successive fasi sperimentali di addestramento e valutazione.

La distribuzione delle classi e la conseguente etichettatura all'interno del dataset riflettono la frequenza e la durata di utilizzo degli strumenti nel contesto chirurgico. In particolare, il numero di esempi associati a ciascuna classe risulta proporzionale al tempo di permanenza degli strumenti nel campo visivo durante le procedure laparoscopiche.

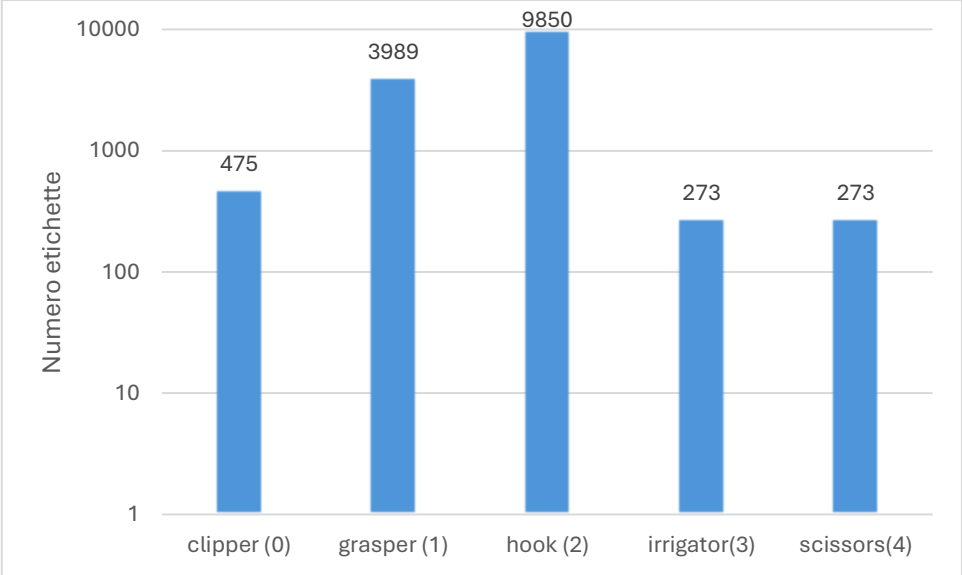


Figura 2.5 Numero di etichette rilevate per classe

Tale scelta consente di mantenere una rappresentazione realistica delle condizioni operative, evitando l'introduzione di bilanciamenti artificiali che potrebbero alterare la distribuzione naturale dei dati. Di conseguenza, il dataset rispecchia fedelmente la dinamica intraoperatoria, favorendo l'addestramento di modelli in grado di operare in scenari reali.

2.2.5. Revisione delle annotazioni ed esportazione dei dati

La qualità delle annotazioni rappresenta un fattore determinante nelle performance dei modelli di object detection, in quanto le reti neurali apprendono direttamente dai dati forniti in fase di addestramento [10,20]. Errori sistematici nella definizione delle classi o nella localizzazione spaziale degli oggetti possono tradursi in una riduzione significativa delle metriche di valutazione e compromettere la capacità di generalizzazione del modello. Per tale motivo, a seguito dell'annotazione si è passati ad una fase di controllo qualità e revisione. Al termine della prima fase di etichettatura, ciascun frame annotato è stato sottoposto a una revisione manuale finalizzata a verificare:

- la correttezza dell'assegnazione della classe;
- la coerenza della nomenclatura utilizzata;
- la precisione della *bounding box* rispetto ai contorni visibili dello strumento;
- l'assenza di oggetti non annotati in presenza di strumenti chiaramente identificabili.

La revisione ha consentito di individuare errori di distrazione, *bounding box* eccessivamente ampie o troppo ristrette, nonché eventuali omissioni. Questo processo ha contribuito a ridurre il rumore nel dataset e a migliorare la qualità complessiva delle etichette.

Il rumore nelle etichette può derivare da errori di classificazione, da *bounding box* inaccurati o da annotazioni mancanti. In ambito laparoscopico, tali problematiche sono accentuate dalla presenza di occlusioni, riflessi e immagini parzialmente sfocate. Per mitigare tali effetti, sono state adottate le seguenti strategie:

- esclusione dei frame in cui lo strumento non risultava chiaramente identificabile;
- rimozione o correzione di bounding box non aderenti ai contorni dell'oggetto;
- uniformità dei criteri di annotazione nei casi limite.

Queste operazioni hanno contribuito a migliorare la qualità statistica del dataset, riducendo la probabilità che il modello apprenda pattern erronei o non rappresentativi.

La relazione tra qualità delle annotazioni e performance del modello è diretta e documentata in letteratura: bounding box imprecise possono abbassare l'*Intersection over Union (IoU)* medio, mentre errori di classificazione incidono negativamente sulla *Mean Average Precision (mAP)*. Un dataset preciso consente di attribuire le variazioni di performance del modello alle scelte architetturali o ai parametri di addestramento, piuttosto che a difetti strutturali nei dati di input.

Una volta completata la fase di annotazione, il dataset è stato esportato dalla piattaforma Roboflow in un formato compatibile con il framework di addestramento. Questa fase rappresenta il punto di transizione tra la costruzione del dataset e lo sviluppo del modello di *object detection*. Le piattaforme di annotazione per *computer vision* consentono generalmente l'esportazione in diversi formati standardizzati, tra cui:

- YOLO format, basato su file di testo contenenti coordinate normalizzate delle bounding box;
- COCO format, strutturato in file JSON con annotazioni dettagliate e metadati;
- Pascal VOC format, basato su file XML associati a ciascuna immagine.

Il dataset è stato esportato dalla piattaforma Roboflow nel formato YOLO, scelto per la compatibilità con la rete neurale YOLOv26. Ogni immagine del dataset è associata a un file di testo .txt contenente le annotazioni relative agli strumenti presenti nella scena. Nel formato YOLO, ogni riga del file .txt corrisponde a una singola istanza di oggetto. All'interno del file vengono riportati, per ogni oggetto annotato:

• identificativo numerico della classe associata all'etichetta

• coordinate x e y del centro della bounding box

• dimensioni della box, espresse come larghezza h e altezza w.

Le coordinate e le dimensioni sono normalizzate rispetto alla dimensione complessiva dell'immagine, assumendo valori compresi tra 0 e 1. Ogni riga del file .txt corrisponde quindi a un oggetto annotato presente nel frame considerato. Questa rappresentazione consente al modello di *object detection* di interpretare la posizione e le dimensioni degli strumenti indipendentemente dalla risoluzione assoluta delle immagini, semplificando la gestione dei dati e l'integrazione nella pipeline di training.

Per garantire una valutazione accurata delle prestazioni del modello, il dataset è stato diviso in tre subset:

- Training (70%): utilizzato per l'addestramento del modello;
- Validation (20%): utilizzato per ottimizzare i parametri e monitorare il rischio di *overfitting* durante il training;
- Test (10%): utilizzato esclusivamente per la valutazione finale delle prestazioni del modello, assicurando una stima oggettiva della capacità di generalizzazione.

La suddivisione è stata eseguita in maniera randomizzata e stratificata, in modo da preservare la distribuzione delle classi di strumenti in ciascun subset, evitando squilibri che potessero influire negativamente sulla capacità predittiva del modello. Questa struttura garantisce riproducibilità e consente confronti affidabili tra differenti esperimenti.

Figura 2.6 Divisione del dataset in training, validation e test set

3. Addestramento delle reti YOLO

Le ultime innovazioni nel *deep learning*, in particolare nelle reti neurali convoluzionali (CNN) per il rilevamento degli oggetti, hanno reso possibile il riconoscimento e il monitoraggio automatico degli strumenti laparoscopici. Questo rappresenta un punto cruciale, poiché consente di raccogliere informazioni essenziali sul loro comportamento durante gli interventi, come la posizione, i movimenti e le interazioni con i tessuti. Tra le varie architetture, i modelli YOLO (*You Only Look Once*) sono diventati tra i più efficaci per il rilevamento in tempo reale grazie alla loro capacità di coniugare precisione e velocità [15,22]. Queste caratteristiche sono particolarmente importanti in chirurgia, dove è fondamentale un'elaborazione rapida delle immagini per un eventuale supporto intraoperatorio.

Nel presente lavoro vengono implementati due modelli di *object detection* basati sull'architettura di YOLOv8 e YOLOv26, addestrati specificamente per il riconoscimento di strumenti laparoscopici all'interno di sequenze video riguardanti interventi di colecistectomia. I modelli sono stati sviluppati utilizzando la piattaforma *Ultralytics*, con un framework basato su *PyTorch* che fornisce strumenti avanzati per l'addestramento, la validazione e l'inferenza di modelli appartenenti alla famiglia YOLO [7].

Il dataset utilizzato per l'addestramento e la validazione dei modelli, descritto nel capitolo precedente, è stato costruito e annotato appositamente per riconoscere le diverse tipologie di strumenti chirurgici presenti nelle immagini laparoscopiche. A partire da tali dati, è stata implementata una pipeline completa che comprende le fasi di configurazione del modello, addestramento della rete neurale, inferenza sulle sequenze video e integrazione con un sistema di tracking degli strumenti nei frame consecutivi.

L'obiettivo di questo capitolo è quindi descrivere nel dettaglio il processo di implementazione della rete neurale e della pipeline di analisi sviluppata. In particolare, verranno illustrate l'architettura del modello utilizzato, la configurazione dell'ambiente di sviluppo sulla piattaforma *Ultralytics*, la procedura di addestramento della rete e l'integrazione con un algoritmo di tracking per il tracciamento temporale degli strumenti laparoscopici. Il modello è stato addestrato per eseguire il rilevamento degli strumenti

mediante la predizione di *bounding boxes* e delle relative classi, consentendo così di localizzare con precisione gli oggetti di interesse all'interno delle immagini.

Oltre alla fase di detection, il modello, attraverso un sistema di *tracking*, deve seguire la posizione degli strumenti nel tempo, associando le rilevazioni ottenute nei frame consecutivi delle sequenze video. Tale funzionalità consente di analizzare il movimento degli strumenti durante l'intervento chirurgico e rappresenta un elemento fondamentale per lo sviluppo di sistemi avanzati di analisi delle procedure laparoscopiche.

A seguito dell'analisi parallela tra le due versioni della rete, verrà scelta quella che otterrà prestazioni migliori, valutate sia dal punto di vista qualitativo che quantitativo.

3.1. Fondamenti dell'object detection

Il problema *dell'object detection* consiste nell'identificazione e nella localizzazione degli oggetti presenti all'interno di un'immagine. A differenza dei problemi di classificazione, nei quali l'obiettivo è assegnare un'etichetta all'intera immagine, *l'object detection* richiede anche la determinazione della posizione degli oggetti all'interno della scena. Tale posizione viene generalmente rappresentata mediante *bounding boxes*, ovvero rettangoli che delimitano l'area dell'immagine contenente l'oggetto di interesse. Gli algoritmi di *object detection* possono essere suddivisi principalmente in due categorie: *two-stage detectors* e *one-stage detectors.*

Nei metodi *two-stage* il processo di rilevamento è suddiviso in due fasi distinte. Nella prima fase viene generato un insieme di regioni candidate che potrebbero contenere oggetti, denominate *region proposals*. Nella seconda fase tali regioni vengono analizzate da un classificatore che determina la classe dell'oggetto e raffina la posizione della *bounding box*. Un esempio rappresentativo di questa categoria è Faster R-CNN, nel quale un modulo denominato *Region Proposal Network* (RPN) genera inizialmente le possibili

regioni contenenti oggetti che vengono successivamente analizzate da una rete di classificazione [16].

Gli approcci *one-stage*, invece, eliminano la fase esplicita di generazione delle regioni candidate e formulano il problema *dell'object detection* come un problema di regressione diretta. In questo caso la rete neurale predice direttamente, a partire dall'immagine di input, le coordinate delle *bounding boxes* e le probabilità di appartenenza alle diverse classi. I modelli appartenenti alla famiglia YOLO rientrano in questa categoria e sono progettati per eseguire il rilevamento degli oggetti in un'unica fase [15]. In questo paradigma, la rete neurale apprende simultaneamente a localizzare e classificare gli oggetti presenti nella scena attraverso un'unica architettura *end-to-end*, riducendo significativamente la complessità computazionale rispetto agli approcci *two-stage.* Nel dettaglio, l'immagine di input viene inizialmente processata da un *backbone* convoluzionale, il cui compito è estrarre una rappresentazione gerarchica delle feature visive. Tali feature vengono successivamente aggregate attraverso moduli di fusione multi-scala, spesso basati su architetture come la *Feature Pyramid Network* o la *Path Aggregation Network*, che consentono di combinare informazioni provenienti da diversi livelli di profondità della rete. I moduli migliorano la capacità del modello di rilevare oggetti caratterizzati da scale differenti [9].

La fase finale dell'architettura è rappresentata dal *detection head*, che produce direttamente, per ciascuna posizione delle *feature maps*, un insieme di predizioni costituite dalle coordinate della *bounding box*, da un punteggio di *objectness*[3] e dalle probabilità di appartenenza alle diverse classi. In molte implementazioni moderne della famiglia YOLO il *detection head* è strutturato in maniera *decoupled*, ovvero con rami separati per la classificazione e per la regressione delle *bounding boxes*. Questa scelta architetturale consente di disaccoppiare l'apprendimento delle informazioni semantiche da quello delle informazioni geometriche, migliorando la stabilità dell'ottimizzazione e le prestazioni complessive del modello. Sebbene il paradigma YOLO appartenga formalmente alla categoria dei detector *one-stage*, alcune delle sue componenti svolgono funzioni analoghe a quelle presenti nei metodi *two-stage*. In particolare, la generazione

[3] Punteggio che indica la probabilità che una regione dell'immagine contenga un oggetto, indipendentemente dalla sua classe.

di molteplici *bounding boxes* candidate per ciascuna posizione della *feature map* può essere interpretata come una forma implicita di generazione di *region proposals*. Successivamente, un algoritmo di post-processing come *Non-Maximum Suppression* viene utilizzato per eliminare le predizioni ridondanti e mantenere solamente le *bounding boxes* con il punteggio più elevato [15]. Grazie a questa formulazione unificata, i modelli YOLO raggiungono prestazioni computazionali particolarmente elevate, rendendo possibile l'esecuzione della *detection* in tempo reale anche su *hardware* con risorse limitate. L'approccio *one-stage* offre diversi vantaggi, in particolare nelle applicazioni che richiedono elaborazione in tempo reale. Eliminando la fase di generazione delle regioni candidate, l'intero processo di rilevamento può essere eseguito mediante un'unica rete neurale convoluzionale, riducendo il numero di operazioni necessarie per produrre le predizioni finali. Questo consente di ottenere velocità di inferenza significativamente più elevate rispetto ai metodi *two-stage*, rendendo tali modelli particolarmente adatti all'elaborazione di sequenze video. Nel contesto della chirurgia laparoscopica, la capacità di elaborare i frame video con una latenza ridotta rappresenta un requisito fondamentale per l'integrazione del sistema in applicazioni di supporto intraoperatorio [22].

3.2. Evoluzione dei modelli YOLO e scelta dell'architettura

La famiglia di modelli YOLO rappresenta uno dei paradigmi più diffusi per il problema *dell'object detection* in tempo reale. Nel corso degli anni, numerose versioni di questa architettura sono state sviluppate con l'obiettivo di migliorare il compromesso tra accuratezza del rilevamento e velocità di inferenza [15,23].

La prima versione, YOLOv1, ha introdotto l'idea di formulare il problema dell*'object detection* come un unico problema di regressione da immagine a *bounding boxes* e probabilità di classe [24]. In questo approccio, l'immagine viene suddivisa in una griglia e ciascuna cella della griglia è responsabile della predizione di un numero limitato di bounding boxes e delle relative probabilità di classe. Sebbene questo modello abbia dimostrato prestazioni molto elevate in termini di velocità, presentava alcune limitazioni nella rilevazione di oggetti piccoli e nella gestione di oggetti vicini tra loro.

Un miglioramento significativo è stato introdotto con YOLOv2, che ha integrato l'utilizzo delle anchor boxes e tecniche di normalizzazione più avanzate [15]. Ciò ha permesso al modello di predire *bounding boxes* di diverse proporzioni, migliorando la capacità di adattarsi alla variabilità delle dimensioni degli oggetti presenti nelle immagini.

Successivamente, YOLOv3 ha introdotto il meccanismo di rilevamento multi-scala, che consente al modello di effettuare predizioni su *feature maps* con diverse risoluzioni spaziali [25]. Questa innovazione ha migliorato significativamente la capacità di individuare oggetti di piccole dimensioni, sfruttando informazioni provenienti da diversi livelli gerarchici della rete.

Le versioni successive, tra cui YOLOv4 e YOLOv5, hanno introdotto ulteriori miglioramenti architetturali e strategie di addestramento più sofisticate. Tra le principali innovazioni si possono citare l'adozione di *backbone* più efficienti, l'utilizzo di tecniche avanzate di *data augmentation* e l'introduzione di moduli di aggregazione delle feature più efficaci. Queste evoluzioni hanno contribuito ad aumentare la precisione del modello mantenendo al contempo elevata la velocità di elaborazione [26].

Le architetture più recenti, come YOLOv6 e YOLOv7, hanno ulteriormente migliorato la progettazione dei moduli interni della rete. In particolare, sono stati introdotti *detection head decoupled*, che separano il processo di classificazione da quello di regressione delle *bounding boxes*, *backbone* ottimizzati per l'estrazione delle feature e meccanismi di fusione multi-scala più efficaci. Queste architetture sono progettate per massimizzare le prestazioni mantenendo una complessità computazionale relativamente contenuta [23].

Nel presente lavoro sono state utilizzate due versioni differenti della famiglia YOLO, ovvero la versione YOLOv8 e YOLOv26. Dal punto di vista architetturale, le differenze tra le due evoluzioni possono essere ricondotte a una serie di evoluzioni progettuali che riguardano principalmente la gestione delle *feature* multi-scala, i meccanismi di aggregazione delle informazioni e le strategie di regressione delle *bounding box*.

YOLOv8 si basa su un'architettura ormai consolidata che adotta una struttura *backbone–neck–head*, con un *backbone* ottimizzato per l'estrazione gerarchica delle feature, un *neck* di tipo PAN-FPN per la fusione multi-scala e una *detection head decoupled*, in cui i rami di classificazione e regressione sono separati per migliorare la stabilità dell'apprendimento. Inoltre, YOLOv8 utilizza un approccio *anchor-free* e integra il *Distribution Focal Loss (DFL*) per affinare la regressione delle *bounding box*, migliorando la precisione nella localizzazione.

YOLOv26 introduce ulteriori miglioramenti su questa base, intervenendo in modo più incisivo sulla rappresentazione e propagazione delle feature. In particolare, l'architettura risulta ottimizzata nella fase di *feature aggregation,* con meccanismi più efficaci di fusione delle informazioni provenienti da diversi livelli della rete, che consentono una migliore preservazione dei dettagli spaziali fini. Questo aspetto è particolarmente rilevante per la rilevazione di oggetti sottili e allungati, come gli strumenti laparoscopici. Inoltre, YOLOv26 adotta strategie più avanzate nella regressione delle *bounding box*, verosimilmente attraverso un utilizzo più raffinato del Distribution Focal Loss e di schemi di ottimizzazione che migliorano la convergenza e la qualità geometrica delle predizioni. Un ulteriore elemento distintivo riguarda la maggiore efficienza nella gestione del *trade-off* tra capacità rappresentativa e complessità computazionale. YOLOv26, pur mantenendo una struttura leggera nella variante "n" (nano), riesce a sfruttare in modo più efficace le feature estratte, migliorando la qualità delle predizioni senza introdurre un

aumento significativo del costo computazionale. Questo suggerisce un'evoluzione nelle scelte progettuali dei blocchi convoluzionali e nei pattern di connessione interna, orientata a massimizzare l'efficacia informativa delle feature maps.

Nel complesso, YOLOv26 può essere interpretato come un'evoluzione di YOLOv8 in cui le principali innovazioni non risiedono tanto in un cambiamento radicale della macro-architettura, quanto in un affinamento dei meccanismi interni di estrazione, fusione e regressione delle informazioni. Tali miglioramenti si traducono in una maggiore capacità del modello di rappresentare oggetti complessi e difficili, garantendo prestazioni superiori soprattutto in scenari applicativi ad alta complessità visiva, come quello laparoscopico.

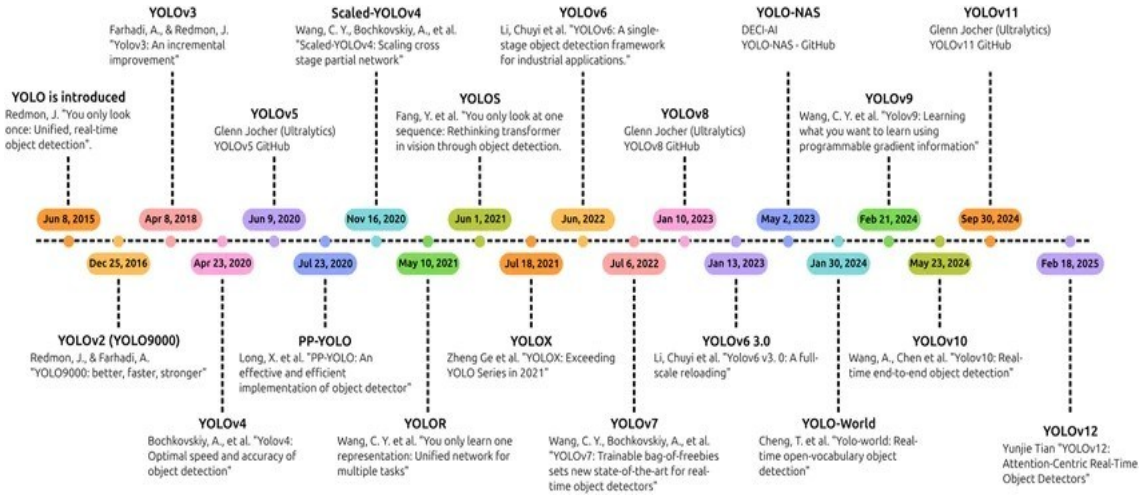


Figura 3.1 Evoluzione delle reti YOLO dal 2015 al 2025

3.3. Architettura delle reti e implementazione dei modelli YOLOv8 e YOLOv26

Nei seguenti paragrafi viene illustrata l'architettura delle reti YOLO, con particolare riferimento ai due modelli adottati nel presente lavoro. Le architetture YOLO appartengono al paradigma *one-stage* per il problema *dell'object detection*, che consente di eseguire simultaneamente la localizzazione e la classificazione degli oggetti mediante un'unica rete neurale convoluzionale *end-to-end*, riducendo la complessità computazionale e permettendo prestazioni compatibili con applicazioni real-time. L'architettura del modello segue una struttura modulare tipica delle reti YOLO moderne ed è composta da tre componenti principali:

- il *backbone*, responsabile dell'estrazione delle feature dall'immagine di input;
- il *neck*, che consente la fusione delle informazioni provenienti da diversi livelli della rete;
- la *detection head*, che produce le predizioni finali relative agli oggetti presenti nell'immagine.

Ciascuna di queste componenti svolge un ruolo specifico nel processo di elaborazione dell'immagine e nella generazione delle predizioni. Nei paragrafi successivi verranno analizzate nel dettaglio le caratteristiche architetturali e le specificità implementative della famiglia di reti con un particolare focus sulle reti YOLOv8 e YOLOv26.

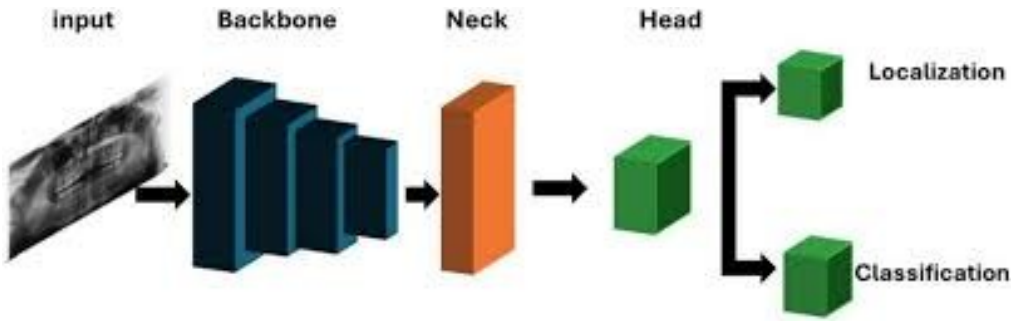


Figura 3.2 Architettura generica di una rete YOLO

.

3.3.1. Backbone

Il *backbone* costituisce la prima componente della rete neurale ed è responsabile dell'estrazione delle feature visive dall'immagine di input. L'obiettivo di questa fase è trasformare l'immagine grezza in un insieme di *feature maps* gerarchiche che catturino progressivamente informazioni sempre più astratte e semanticamente significative [14,18].

Esso è generalmente costituito da una sequenza di blocchi convoluzionali organizzati in più livelli di profondità. Ogni blocco è tipicamente composto da tre operazioni principali in una configurazione comunemente indicata come blocco Conv–BN–Activation:

- Convoluzione
- normalizzazione (Batch Normalization)
- funzione di attivazione non lineare

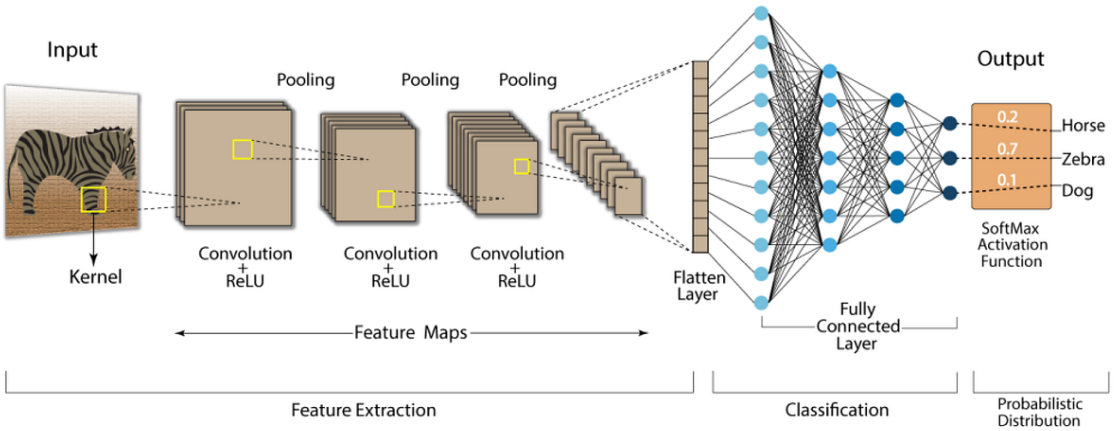


Figura 3.3 Schema del blocco Conv–BN–Activation: sequenza di convoluzione, normalizzazione e funzione di attivazione utilizzata nelle reti neurali convoluzionali.

La convoluzione rappresenta l'operazione fondamentale attraverso cui vengono estratte le caratteristiche locali dell'immagine [18]. Essa può essere espressa come:

$$y(i,j) = m\sum n\sum x(i+m, j+n)w(m,n) \quad (1)$$

dove $x(i+m, j+n)$ rappresenta il valore dell'immagine di input nella posizione $(i+m, j+n)$, $w(m,n)$indica i pesi del *kernel* convoluzionale, mentre m e n sono gli indici che scorrono sulle dimensioni del filtro. Il termine $y(i,j)$rappresenta il valore della

feature map in uscita nella posizione (i, j). In tale processo le maschere hanno tipicamente dimensione 3×3 o 1×1; vengono applicate localmente alle diverse regioni dell'immagine, generando nuove rappresentazioni. Nei primi livelli della rete, i filtri risultano sensibili a caratteristiche a basso livello, come bordi e texture, mentre nei livelli più profondi le feature diventano progressivamente più complesse, rappresentando strutture semantiche più articolate [14].

A valle dell'operazione di convoluzione viene applicata la *Batch Normalization* (BN), che stabilizza le attivazioni normalizzandole rispetto alla media e alla varianza del *batch*[4], mantenendo al contempo flessibilità grazie a parametri apprendibili [27]. In particolare, data un'attivazione x, la *Batch Normalization* può essere espressa come:

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{2}$$

dove μ_Be σ_B^2 rappresentano rispettivamente la media e la varianza calcolate sul *batch*, mentre ϵ è un termine di stabilizzazione numerica. Il valore normalizzato viene successivamente scalato e traslato tramite parametri apprendibili γe β

$$y = \gamma x\hat{} + \beta$$

Questa operazione contribuisce a migliorare la stabilità numerica del training, favorendo una più rapida convergenza e una minore sensibilità alla scelta degli iperparametri.

Successivamente, viene applicata una funzione di attivazione non lineare, come ReLU, Leaky ReLU o SiLU, che introduce non linearità nel modello e consente alla rete di apprendere relazioni complesse tra le feature.

Nei *backbone* delle architetture YOLO moderne, le *feature maps* vengono inoltre sottoposte a operazioni di *downsampling*[5], realizzate mediante convoluzioni con stride maggiore di uno oppure operazioni di *pooling*[6] [7]. Questo processo riduce progressivamente la risoluzione spaziale delle *feature maps*, aumentando il *receptive field*

[4] sottoinsieme del dataset utilizzato per aggiornare i pesi della rete durante una singola iterazione di addestramento
[5] Operazione in cui viene ridotta la risoluzione spaziale delle feature maps, mantenendo al contempo le informazioni più rilevanti
[6] Operazione che riduce la dimensione spaziale delle feature maps, aggregando le informazioni contenute in regioni locali

dei neuroni nei livelli più profondi e consentendo alla rete di catturare informazioni contestuali su regioni sempre più ampie dell'immagine [18].

Nel modello YOLOv26, il *backbone* è progettato per garantire un'elevata efficienza computazionale mantenendo al contempo una buona capacità di rappresentazione delle caratteristiche visive. L'architettura segue una struttura gerarchica nella quale, all'aumentare della profondità, la risoluzione spaziale delle *feature maps* diminuisce mentre aumenta il numero di canali, consentendo di ottenere rappresentazioni sempre più astratte e semanticamente informative [14]. Una caratteristica distintiva del *backbone* di YOLOv26 è l'impiego di moduli basati sul principio delle *Cross Stage Partial Connections* (CSP), introdotte nell'architettura CSPNet e implementate nelle moderne varianti YOLO attraverso blocchi denominati C2f [7,28]. In tali moduli, il flusso delle feature maps viene suddiviso in due percorsi paralleli: una parte delle feature viene elaborata attraverso una serie di blocchi convoluzionali, mentre un'altra parte viene propagata direttamente verso l'uscita dello stadio senza ulteriori trasformazioni. I due flussi vengono quindi ricombinati mediante concatenazione lungo la dimensione dei canali. Questa struttura presenta diversi vantaggi. In primo luogo, la presenza di un percorso diretto facilita la propagazione del gradiente durante l'addestramento, migliorando la stabilità dell'ottimizzazione nelle reti profonde [28] .

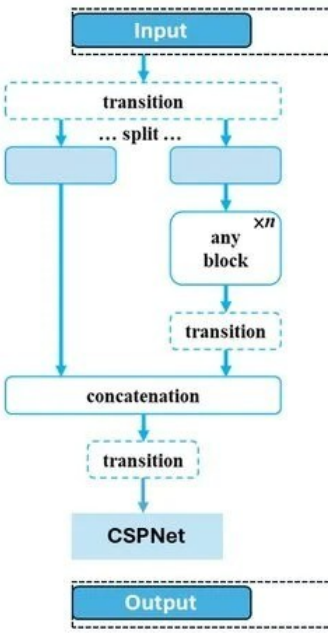


Figura 3.4 Architettura di una CSPnet

In secondo luogo, la suddivisione del flusso informativo consente di ridurre la ridondanza computazionale, limitando il numero complessivo di operazioni richieste dal modello. Infine, la combinazione delle feature provenienti dai due percorsi permette di preservare simultaneamente informazioni originali e trasformate, migliorando la capacità rappresentativa complessiva del *backbone*.

Il risultato finale del *backbone* è un insieme di *feature maps* a diverse scale e livelli di profondità, che forniscono una rappresentazione compatta ma ricca di informazioni della scena. Tali rappresentazioni costituiscono l'input per il modulo di fusione multi-scala presente nel neck della rete.

A livello architetturale, il backbone di YOLOv26 si colloca in una linea evolutiva rispetto a quello adottato in YOLOv8, condividendone i principi fondamentali ma introducendo alcune ottimizzazioni strutturali. In YOLOv8, il *backbone* è progettato secondo una struttura gerarchica modulare basata su blocchi *Conv–BN–Activation* e sull'impiego dei moduli C2f, derivati dal paradigma delle *Cross Stage Partial connections* (CSP), con l'obiettivo di bilanciare capacità rappresentativa ed efficienza computazionale [7].

Nel modello YOLOv26, tali principi vengono ulteriormente raffinati mediante una riorganizzazione più efficiente del flusso informativo tra gli stadi della rete. In particolare, pur mantenendo l'uso di connessioni parziali tra i layer, YOLOv26 introduce una gestione più efficiente della suddivisione delle feature maps nei percorsi paralleli, migliorando il riutilizzo delle informazioni e riducendo la ridondanza computazionale. Questo approccio consente una propagazione del gradiente più stabile durante l'addestramento e contribuisce a una maggiore robustezza dell'ottimizzazione nelle reti profonde [28].

Un ulteriore elemento distintivo riguarda la gestione del compromesso tra risoluzione spaziale e profondità dei canali. Entrambe le architetture seguono una struttura piramidale in cui la risoluzione delle feature maps diminuisce progressivamente con la profondità della rete, mentre aumenta il numero di canali. Tuttavia, YOLOv26 adotta una strategia più marcata nell'espansione dei canali nei livelli profondi, favorendo l'apprendimento di rappresentazioni più astratte e semanticamente ricche. Tale scelta permette di migliorare la capacità discriminativa del modello, mantenendo al contempo un'elevata efficienza computazionale.

Nel complesso, YOLOv26 può essere interpretato come un'evoluzione del backbone di YOLOv8, in cui i principi di modularità, gerarchia multi-scala e utilizzo delle CSP connections vengono ulteriormente ottimizzati per ottenere un miglior compromesso tra accuratezza, stabilità del training e costo computazionale. Questa evoluzione riflette una tendenza generale nello sviluppo delle moderne architetture di *object detection*, orientata verso modelli sempre più efficienti e scalabili [14].

3.3.2. Neck

Il *neck* rappresenta il modulo intermedio dell'architettura ed è responsabile della combinazione delle *feature maps* prodotte dal backbone, al fine di costruire rappresentazioni multi-scala utili al processo di rilevamento degli oggetti [28,29]. Nelle architetture della famiglia YOLO, questa componente svolge un ruolo fondamentale nel migliorare la capacità del modello di rilevare oggetti caratterizzati da dimensioni differenti.

Durante il processo di estrazione delle feature, il *backbone* produce rappresentazioni a diverse risoluzioni spaziali: i livelli più profondi generano *feature maps* a bassa risoluzione ma ad elevato contenuto semantico, mentre i livelli più superficiali producono mappe ad alta risoluzione contenenti informazioni più locali e meno astratte [29]. Il neck ha quindi il compito di integrare tali rappresentazioni, combinando simultaneamente informazioni semantiche e dettagli spaziali.

A tal fine, le moderne architetture YOLO adottano meccanismi di fusione multi-scala basati su strutture derivate dalla *Feature Pyramid Network* (FPN) e dalla *Path Aggregation Network* (PAN) [28,29].

Nel caso della FPN, la fusione avviene attraverso un percorso *top-down*, in cui le feature maps provenienti dai livelli più profondi vengono progressivamente aumentate di risoluzione mediante operazioni di *upsampling* e combinate con le feature maps dei livelli precedenti tramite connessioni laterali. Questo processo consente di arricchire le rappresentazioni ad alta risoluzione con informazioni semantiche più profonde [29].

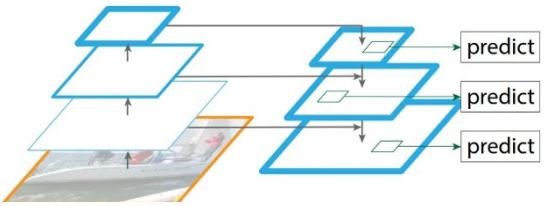


Figura 3.5 Schema della Feature Pyramid Network (FPN)

La *Path Aggregation Network* introduce invece un percorso *bottom-up* aggiuntivo, che migliora la propagazione delle informazioni lungo la gerarchia delle feature. In questo caso, le feature maps vengono ulteriormente aggregate mediante operazioni di *downsampling* e convoluzione, favorendo una più efficace integrazione tra informazioni locali e globali e riducendo la distanza informativa tra i diversi livelli della rete.

Nel modello YOLOv26, il *neck* implementa tali strategie di fusione multi-scala combinando *feature maps* a diverse risoluzioni tramite operazioni di *upsampling*, *downsampling* e concatenazione. Questo processo consente di ottenere rappresentazioni arricchite, nelle quali informazioni spaziali locali e contenuti semantici globali risultano efficacemente integrati. L'architettura produce tipicamente tre livelli di feature maps a risoluzione progressivamente decrescente, ciascuno specializzato nel rilevamento di oggetti appartenenti a differenti scale dimensionali [7]. Le *feature maps* ad alta risoluzione risultano particolarmente efficaci nel rilevamento di oggetti di piccole dimensioni, mentre quelle a risoluzione inferiore consentono di individuare oggetti più grandi. Nel contesto delle immagini laparoscopiche, questa capacità di analisi multi-scala assume un ruolo particolarmente rilevante, poiché gli strumenti chirurgici possono apparire con dimensioni variabili in funzione della distanza dalla telecamera e della prospettiva della scena. L'integrazione di informazioni provenienti da diverse scale consente quindi al modello di mantenere elevate prestazioni anche in presenza di significative variazioni dimensionali.

In YOLOv8, il neck è anch'esso basato su una combinazione di FPN e PAN, progettata per garantire un'efficace fusione multi-scala mantenendo al contempo un'elevata efficienza computazionale. Tuttavia, nel modello YOLOv26 tale struttura viene ulteriormente ottimizzata attraverso una gestione più efficiente delle operazioni di aggregazione e concatenazione delle feature maps.

In particolare, YOLOv26 migliora il flusso informativo tra i diversi livelli della rete, riducendo la perdita di informazioni durante le operazioni di *upsampling* e *downsampling* e favorendo una più efficace integrazione tra dettagli spaziali e contenuto semantico. Questo si traduce in una maggiore qualità delle rappresentazioni multi-scala e, di conseguenza, in un miglioramento delle prestazioni nel rilevamento di oggetti di dimensioni variabili.

Nel complesso, il neck di YOLOv26 può essere interpretato come un'evoluzione di quello adottato in YOLOv8, in cui le strategie di fusione multi-scala vengono ulteriormente raffinate per ottenere un miglior equilibrio tra accuratezza, robustezza e costo computazionale.

3.3.3. Detection head

La *detection head* costituisce il modulo finale dell'architettura ed è responsabile della trasformazione delle *feature maps* multi-scala prodotte dal *neck* nelle predizioni finali relative agli oggetti presenti nell'immagine [15]. Nelle architetture della famiglia YOLO, essa opera direttamente su *feature maps* a diverse risoluzioni spaziali e produce, per ciascuna posizione della griglia, un insieme di parametri che descrivono la presenza e la localizzazione degli oggetti.

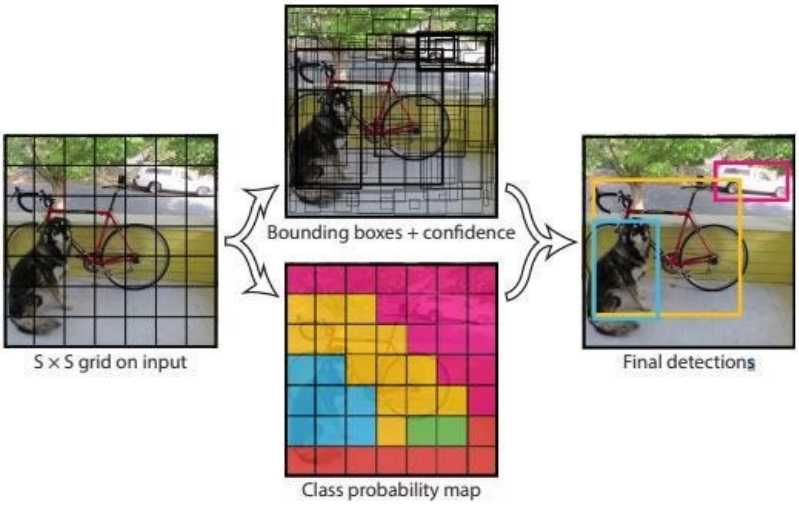


Figura 3.6 Implementazione delle predizioni nella detection head

In particolare, ogni cella della *feature map* viene interpretata come una possibile posizione candidata per la presenza di un oggetto. Per ciascuna posizione, la *detection head* genera una serie di predizioni che includono le coordinate della *bounding box*, un punteggio di *objectness* che rappresenta la probabilità che la regione contenga effettivamente un oggetto, e le probabilità di appartenenza alle diverse classi considerate dal modello.

Dal punto di vista architetturale, la *detection head* è generalmente costituita da una sequenza di *layer* convoluzionali di piccola dimensione, tipicamente con *kernel* 1×1 o 3×3, che trasformano le *feature maps* in vettori di predizione per ciascuna posizione spaziale. Nelle implementazioni moderne, tra cui il modello YOLOv26, la *detection head* è organizzata secondo una struttura *decoupled*, nella quale il ramo di classificazione e quello di regressione delle bounding boxes sono separati (2). Questo approccio consente

di specializzare i due sottoprocessi, migliorando la stabilità dell'addestramento e la qualità delle predizioni.

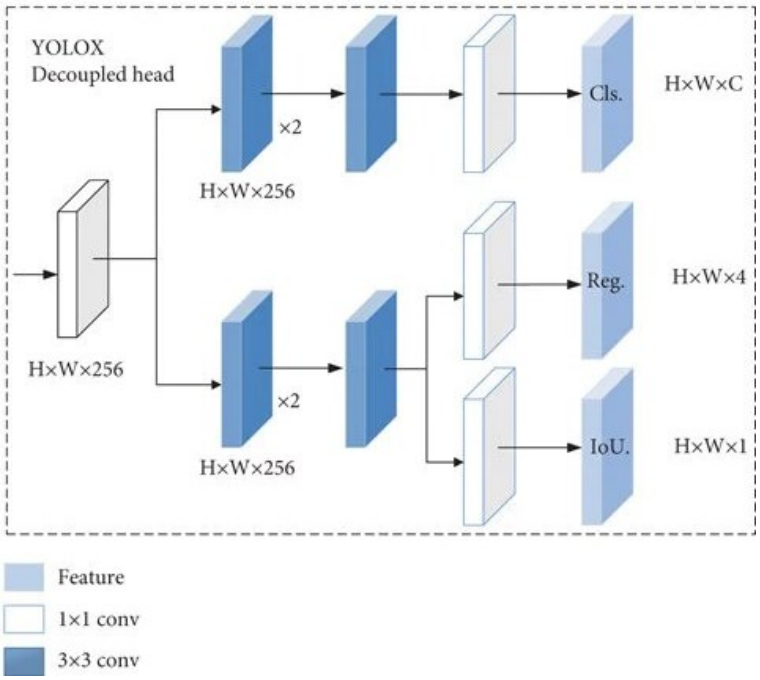


Figura 3.7 Schema della detection head decoupled

Il ramo di regressione ha il compito di stimare i parametri geometrici delle *bounding boxes*, generalmente rappresentati dalle coordinate del centro e dalle dimensioni, che vengono successivamente riportati nello spazio dell'immagine originale tramite opportune operazioni di scalatura. Il ramo di classificazione, invece, produce un vettore di probabilità che indica l'appartenenza dell'oggetto alle diverse classi, generalmente ottenuto tramite funzioni di attivazione come sigmoide o softmax. Un ulteriore elemento prodotto dalla *detection head* è il punteggio di *objectness*, che rappresenta la probabilità che una *bounding box* contenga effettivamente un oggetto indipendentemente dalla classe [15].

Poiché ogni posizione della *feature map* può generare più predizioni candidate, il modello produce tipicamente un numero elevato di *bounding boxes* potenziali. Per ottenere il set finale di oggetti rilevati viene quindi applicato un algoritmo di post-processing come la *Non-Maximum Suppression* (NMS), che elimina le predizioni ridondanti mantenendo solamente quelle con il punteggio di confidenza più elevato e con minore sovrapposizione spaziale.

Figura 3.8 Algoritmo di Non-Max-Suppression

Grazie alla presenza di predizioni effettuate su più scale di *feature maps*, la *detection head* è in grado di rilevare oggetti caratterizzati da dimensioni differenti. Le feature maps ad alta risoluzione risultano particolarmente efficaci per il rilevamento di oggetti di piccole dimensioni, mentre quelle a risoluzione inferiore consentono di individuare oggetti più grandi.

Nel contesto delle immagini laparoscopiche, questa capacità risulta particolarmente rilevante, poiché gli strumenti chirurgici possono apparire con dimensioni variabili in funzione della distanza dalla telecamera e della prospettiva della scena. L'approccio one-stage adottato consente inoltre di mantenere tempi di inferenza compatibili con applicazioni in tempo reale.

In YOLOv8, la *detection head* è anch'essa organizzata secondo una struttura *decoupled*, nella quale i rami di classificazione e regressione sono separati al fine di migliorare la qualità delle predizioni e la stabilità dell'addestramento. Tale configurazione consente una buona specializzazione dei sottocompiti, contribuendo a prestazioni elevate nel rilevamento degli oggetti. Nel modello YOLOv26, questa struttura viene ulteriormente

ottimizzata attraverso una gestione più efficiente del flusso informativo tra i rami della *head* e una migliore integrazione con le *feature maps* provenienti dal *neck*. In particolare, le operazioni di regressione e classificazione risultano maggiormente specializzate, permettendo una stima più accurata delle bounding boxes e una migliore discriminazione tra le classi [7,15].

Nel complesso, la *detection head* di YOLOv26 può essere interpretata come un'evoluzione di quella adottata in YOLOv8, in cui il paradigma *decoupled* viene ulteriormente raffinato per migliorare l'accuratezza delle predizioni mantenendo al contempo un'elevata efficienza computazionale.

3.4. Predizioni della rete

Il meccanismo di predizione nei modelli YOLO si basa su una formulazione unificata del problema dell'*object detection*, in cui localizzazione e classificazione degli oggetti vengono eseguite simultaneamente mediante un'unica rete neurale. In questo approccio, l'immagine di input viene suddivisa in una griglia di celle, ciascuna delle quali è responsabile della predizione degli oggetti presenti nella corrispondente regione dell'immagine. A partire da questa rappresentazione, il modello genera, per ogni cella della griglia, un insieme di predizioni che includono la posizione degli oggetti sotto forma di *bounding boxes*, un punteggio di confidenza associato alla presenza dell'oggetto e le probabilità di appartenenza alle diverse classi. Questa formulazione consente di trasformare il problema del rilevamento in un problema di regressione diretta, permettendo di ottenere prestazioni elevate in termini di velocità di inferenza e rendendo i modelli YOLO particolarmente adatti ad applicazioni in tempo reale.

Nei paragrafi seguenti vengono descritti in dettaglio i principali elementi che caratterizzano il processo di predizione: la rappresentazione delle *bounding boxes*, il

calcolo del *confidence score*, la stima delle probabilità di classe e la funzione di perdita utilizzata durante l'addestramento.

3.4.1. Bounding box

Una *bounding box* rappresenta una regione rettangolare che delimita la posizione di un oggetto all'interno dell'immagine. Generalmente essa viene descritta mediante quattro parametri:

$$(x, y, w, h)$$

dove:

- xe y rappresentano le coordinate del centro della *bounding box*
- w e h rappresentano la larghezza e l'altezza della bounding box.

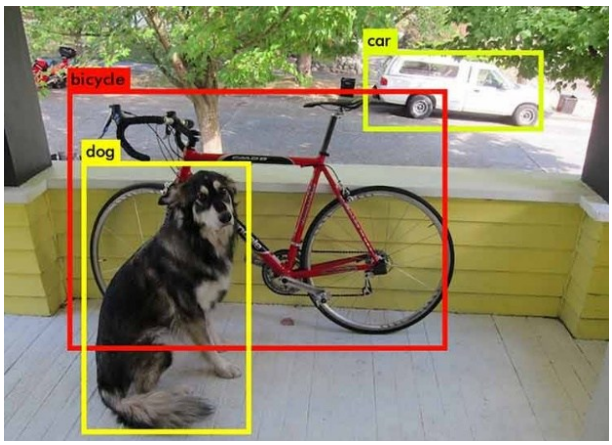


Figura 3.9 Creazione delle bounding boxes

Questi parametri permettono di definire in modo preciso la posizione e la dimensione dell'oggetto rilevato. Nei modelli YOLO, le coordinate della bounding box sono generalmente normalizzate rispetto alle dimensioni dell'immagine, assumendo valori compresi tra 0 e 1. Questo consente al modello di operare indipendentemente dalla risoluzione dell'immagine e di mantenere una rappresentazione scalabile [15]. Inoltre, le coordinate della bounding box sono predette relativamente alla cella della griglia a cui

appartiene l'oggetto. In particolare, x e y rappresentano la posizione del centro della bounding box rispetto alla cella della griglia, mentre w e $e\ h$ descrivono le dimensioni della regione rilevata.

3.4.2. Confidence score

Il *confidence score* rappresenta una misura della probabilità che una determinata *bounding box* contenga effettivamente un oggetto e consente di avere informazioni sulla qualità della sua localizzazione.

Nei modelli YOLO, questo valore è legato sia alla probabilità di presenza dell'oggetto all'interno della *bounding box* sia alla precisione con cui essa approssima la posizione reale dell'oggetto (3). In particolare, il *confidence score* può essere espresso come:

$$\text{confidence} = P(\text{oggetto}) \cdot IoU \tag{3}$$

dove $P(\text{oggetto})$rappresenta la probabilità che la bounding box contenga un oggetto, mentre IoU misura la qualità della localizzazione.

Per quantificare la qualità della localizzazione viene utilizzata la metrica *Intersection over Union* (IoU), che misura il grado di sovrapposizione tra la bounding box predetta e quella reale. Essa è definita come:

$$IoU = \frac{\text{Area di intersezione}}{\text{Area di unione}} \tag{4}$$

dove l'area di intersezione rappresenta la regione comune alle due *bounding boxes*, mentre l'area di unione corrisponde all'area complessiva coperta da entrambe.

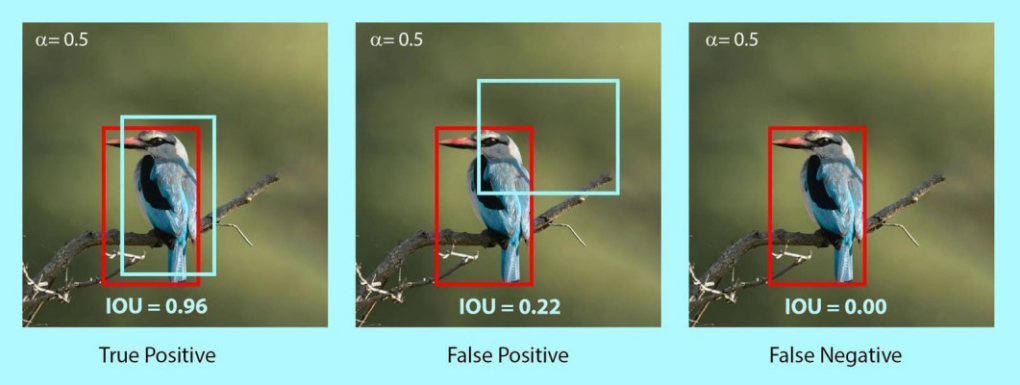


Figura 3.10 Algoritmo di Intersection over Union (IoU)

Il valore di IoU è compreso tra 0 e 1: valori prossimi a 1 indicano una forte sovrapposizione e quindi una localizzazione accurata, mentre valori bassi indicano una scarsa corrispondenza tra predizione e ground truth. In fase di valutazione e *post-processing*, la metrica IoU viene utilizzata anche per determinare la qualità delle predizioni e per applicare tecniche come la *Non-Maximum Suppression*, che elimina *bounding boxes* sovrapposte mantenendo quelle con maggiore confidenza.

3.4.3. Class probability

La rete è in grado di predire anche l'appartenenza degli oggetti alle diverse classi. Per ciascuna *bounding box*, il modello produce un vettore di probabilità che descrive la distribuzione di probabilità sulle classi considerate. Questo vettore è ottenuto come output del ramo di classificazione della *detection head*, generalmente attraverso una funzione di attivazione (ad esempio sigmoide o softmax), che consente di interpretare i valori come probabilità associate a ciascuna classe. Il punteggio finale associato a una predizione può essere espresso come:

$$Score = P(\text{object}) \times P(\text{class} \mid \text{object}) \tag{5}$$

dove $P(\text{object})$ rappresenta la probabilità che nella bounding box sia presente un oggetto, mentre $P(\text{class} \mid \text{object})$ rappresenta la probabilità condizionata che tale oggetto appartenga a una specifica classe. Questo meccanismo consente al modello di combinare

l'informazione relativa alla presenza dell'oggetto con la sua classificazione, producendo una misura complessiva della confidenza della predizione [15].

3.4.4. Loss function

Durante la fase di addestramento, il modello viene ottimizzato mediante una funzione di perdita (*loss function*), che misura la discrepanza tra le predizioni della rete e i valori reali presenti nel dataset annotato. L'obiettivo dell'ottimizzazione è quello di aggiornare i parametri della rete in modo da ridurre progressivamente tale errore attraverso tecniche di *backpropagation*. Nei modelli di *object detection* della famiglia YOLO, la funzione di *loss* è generalmente composta da più termini, combinati tra loro come una somma pesata, ciascuno dei quali contribuisce all'apprendimento di un aspetto specifico del problema:

- *loss* di localizzazione, che penalizza errori nella stima delle coordinate delle bounding boxes, migliorando la precisione spaziale delle predizioni
- *loss* di classificazione, che misura l'errore nella predizione delle classi degli oggetti, favorendo una corretta distinzione tra le diverse categorie
- *loss* di *objectness*, che valuta la correttezza della predizione relativa alla presenza di un oggetto all'interno della bounding box

L'ottimizzazione congiunta di queste componenti consente alla rete di apprendere simultaneamente a localizzare e classificare gli oggetti presenti nelle immagini. Questo approccio è alla base del funzionamento dei modelli YOLO, nei quali il problema dell'*object detection* viene formulato come un task di apprendimento unificato.

3.5. Implementazione sulla piattaforma Ultralytics

L'implementazione del modello di *object detection* utilizzato nel presente lavoro è stata realizzata mediante il framework *open-source* Ultralytics [7], una piattaforma progettata per facilitare lo sviluppo, l'addestramento e l'utilizzo dei modelli appartenenti alla famiglia YOLO. Tale framework consente di semplificare l'intera pipeline di deep learning, integrando strumenti per la gestione del dataset, l'addestramento del modello e il monitoraggio delle prestazioni, permettendo una rapida configurazione e un'elevata modularità. La struttura della libreria è organizzata in moduli distinti dedicati alla gestione delle architetture dei modelli, al caricamento e *preprocessing* dei dati, al processo di training e alla fase di inferenza, garantendo una chiara separazione delle diverse componenti del sistema e una maggiore flessibilità nell'adattamento a specifici contesti applicativi. Il framework Ultralytics è sviluppato su PyTorch, una libreria *open source* ampiamente utilizzata nel campo del *deep learning*. PyTorch si basa su un paradigma di grafi computazionali dinamici, ossia strutture che rappresentano le operazioni matematiche (come somme, prodotti e trasformazioni) eseguite sui dati. Il termine dinamici indica che tali grafi vengono costruiti durante l'esecuzione del modello, passo dopo passo, e non definiti completamente in anticipo. Questo approccio consente una maggiore flessibilità e facilita la modifica e l'analisi del comportamento del modello.

Un elemento centrale di PyTorch è il meccanismo di *automatic differentiation*, implementato tramite il modulo *autograd*. Per differenziazione automatica si intende la capacità del sistema di calcolare automaticamente le derivate (o gradienti) di una funzione rispetto alle sue variabili. In questo contesto, il termine gradiente indica quanto una piccola variazione dei parametri del modello influisce sul valore della funzione di errore (detta anche funzione di perdita). Il modulo *autograd* tiene traccia di tutte le operazioni eseguite sui dati e costruisce implicitamente il grafo computazionale necessario per calcolare tali gradienti. I gradienti così ottenuti vengono utilizzati nel processo di *backpropagation*, algoritmo che permette di aggiornare i parametri della rete neurale partendo dall'errore commesso in uscita e propagandolo all'indietro attraverso i vari livelli della rete. In questo modo, il modello è in grado di "imparare" dai dati.

Infine, l'aggiornamento dei parametri avviene tramite algoritmi di ottimizzazione, come ad esempio lo *Stochastic Gradient Descent (SGD)* o *Adam*, che modificano iterativamente

i pesi della rete con l'obiettivo di ridurre la funzione di perdita e migliorare progressivamente le prestazioni del modello [14]. L'integrazione con PyTorch consente inoltre di sfruttare l'accelerazione hardware tramite GPU, riducendo significativamente i tempi di addestramento e rendendo possibile l'esecuzione di modelli complessi su dataset di dimensioni rilevanti.

La configurazione dell'ambiente di sviluppo ha previsto l'utilizzo di un sistema dotato di unità di elaborazione grafica (GPU), affiancato da una CPU e da memoria RAM adeguata alla gestione del dataset e delle operazioni di training. In particolare, la GPU è stata impiegata per l'esecuzione delle operazioni computazionali intensive, come le convoluzioni e l'ottimizzazione dei pesi, mentre la CPU ha gestito il caricamento dei dati e le operazioni ausiliarie. L'utilizzo della GPU ha consentito di ridurre significativamente i tempi di addestramento e di eseguire un numero maggiore di esperimenti.

Il dataset utilizzato per il rilevamento degli strumenti laparoscopici è stato caricato direttamente all'interno del framework Ultralytics, che provvede automaticamente alla gestione della struttura dei dati e alla configurazione del processo di *training*, inclusa la suddivisione tra *training* e *validation set*, il riconoscimento delle classi e la gestione delle annotazioni nel formato YOLO. Il processo di addestramento è stato avviato utilizzando modelli pre-addestrati, sfruttando tecniche di *transfer learning* per migliorare la capacità di generalizzazione del modello e ridurre i tempi di convergenza.

L'ottimizzazione del modello è stata effettuata definendo opportuni iperparametri, tra cui il *learning rate*, che controlla la velocità di aggiornamento dei pesi della rete, il *batch size*, che determina il numero di immagini elaborate per iterazione, il numero di epoche, che rappresenta il numero di passaggi completi sul dataset, e la dimensione delle immagini di input, che influenza la qualità delle feature estratte. La scelta di tali parametri ha un impatto diretto sulle prestazioni del modello e sulla stabilità del processo di apprendimento.

L'utilizzo del framework Ultralytics ha quindi consentito di implementare in modo efficiente un sistema di object detection basato su YOLO, garantendo al contempo semplicità d'uso, flessibilità e compatibilità con le principali tecnologie di deep learning, risultando particolarmente adatto al contesto applicativo considerato.

3.6. Addestramento delle reti e scelta della rete ottima

La definizione di una strategia di training appropriata rappresenta un passaggio fondamentale per garantire la capacità di generalizzazione del modello e prevenire fenomeni di *overfitting* e *underfitting* [14]. Nel presente lavoro, il processo di addestramento è stato strutturato mediante una suddivisione del dataset in tre sottoinsiemi distinti: training, validation e test. Il training set è stato impiegato per l'ottimizzazione dei pesi della rete neurale, consentendo al modello di apprendere le caratteristiche visive degli strumenti laparoscopici a partire dalle immagini annotate. Il validation set è stato utilizzato durante il training per monitorare le prestazioni del modello e valutarne la capacità di generalizzazione su dati non visti, permettendo di individuare eventuali fenomeni di overfitting. Infine, il test set è stato utilizzato esclusivamente nella fase finale di valutazione, al fine di ottenere una stima imparziale delle prestazioni del modello su dati completamente nuovi. Questa suddivisione consente di garantire una valutazione più robusta ed affidabile del modello addestrato. Nel contesto del *deep learning*, una strategia ampiamente adottata consiste nell'utilizzo di modelli pre-addestrati su dataset di grandi dimensioni, come ImageNet o COCO, successivamente adattati a specifici problemi mediante tecniche di fine-tuning [9]. In questo lavoro è stata adottata una strategia di *transfer learning*, nei quali entrambi i modelli sono stati allenati con pesi pre-addestrati e successivamente riaddestrato sul dataset di strumenti laparoscopici. Questo approccio consente di sfruttare feature generiche già apprese su grandi quantità di dati, migliorando la capacità del modello di adattarsi rapidamente a contesti specifici e riducendo significativamente i tempi di addestramento.

3.6.1. Configurazione degli iperparametri

La definizione degli iperparametri di *training* rappresenta un aspetto cruciale per il corretto funzionamento del processo di apprendimento della rete neurale, in quanto tali parametri influenzano direttamente la velocità di convergenza, la stabilità dell'ottimizzazione e le prestazioni finali del modello. Nel presente lavoro, la scelta degli iperparametri è stata la medesima per la versione di YOLOv8 e YOLOv26; l'addestramento dei modelli è stato effettuato utilizzando il *framework* Ultralytics, configurando gli iperparametri in funzione delle caratteristiche del dataset laparoscopico e delle risorse computazionali disponibili. In particolare, il modello è stato addestrato per un totale di 100 epoche, consentendo alla rete di apprendere progressivamente le caratteristiche visive degli strumenti chirurgici, mantenendo un buon equilibrio tra apprendimento e rischio di *overfitting*.

- Il *learning rate* iniziale (lr0 = 0.01**)** rappresenta la velocità con cui i pesi della rete vengono aggiornati durante l'addestramento: valori elevati consentono un apprendimento rapido ma possono causare instabilità, mentre valori più bassi garantiscono una convergenza più stabile ma più lenta. Il parametro lrf = 0.01 introduce un decadimento progressivo del *learning rate*, permettendo aggiornamenti più fini nelle fasi finali del training. Inoltre, l'utilizzo di un periodo di *warmup* (*warmup_epochs* = 3.0**)** consente di inizializzare gradualmente il processo di apprendimento, evitando oscillazioni dei pesi nelle prime iterazioni (2).
- Il *batch size* = 16 indica il numero di immagini elaborate simultaneamente durante ogni iterazione e influisce sulla stima del gradiente: batch più grandi producono aggiornamenti più stabili, mentre batch più piccoli introducono maggiore variabilità ma possono favorire la generalizzazione del modello. La dimensione delle immagini di input (imgsz = 640) influisce direttamente sulla capacità del modello di catturare dettagli spaziali: valori più elevati migliorano la localizzazione degli oggetti, ma aumentano il costo computazionale.
- L'ottimizzazione è stata eseguita con un optimizer automatico, con momentum pari a 0.8, parametro che consente di accelerare la convergenza accumulando

informazioni sugli aggiornamenti precedenti, e weight decay pari a 0.0005, che introduce una regolarizzazione sui pesi per ridurre il rischio di overfitting.

- Sono state inoltre applicate tecniche di data augmentation, tra cui mosaic, flip orizzontale (fliplr = 0.5) e RandAugment, che permettono di aumentare artificialmente la variabilità del dataset, migliorando la capacità del modello di generalizzare su dati non visti.

Parametri	**Valore**	**Descrizione**
Learning rate (lr0)	0.01	Velocità iniziale di aggiornamento dei pesi
Learning rate finale (lrf)	0.01	Decadimento del learning rate durante il training
Momentum	0.8	Stabilizza l'aggiornamento dei pesi
Weight decay	0.0005	Regolarizzazione per ridurre overfitting
Optimizer	Auto	Selezione automatica dell'algoritmo di ottimizzazione
Numero di epoche	100	Numero di iterazioni complete sul dataset
Batch size	16	Numero di immagini per iterazione
Warmup epochs	3.0	Fase iniziale di stabilizzazione del training
Dimensione immagini (imgsz)	640	Risoluzione delle immagini di input
Mosaic	True	Combinazione di più immagini durante il training
Flip orizzontale (fliplr)	0.5	Probabilità di riflessione orizzontale
RandAugment	Attivo	Tecnica automatica di data augmentation
Workers	8	Numero di processi per il caricamento dei dati

Figura 3.11 Scelta degli iperparmetri di training

3.6.2. Configurazione del modello

La configurazione del modello rappresenta una fase fondamentale nel processo di sviluppo di un sistema di *object detection*, poiché determina l'architettura della rete e le modalità di elaborazione delle informazioni visive. Nel presente lavoro sono state utilizzate le cosiddette versioni "nano" (YOLOv8n, YOLOv26n), varianti della famiglia YOLO, selezionate per il loro favorevole compromesso tra accuratezza e costo computazionale (4). La scelta di questo modello è stata guidata dalla necessità di operare su immagini laparoscopiche, caratterizzate da elevata variabilità visiva, presenza di occlusioni e dimensioni variabili degli oggetti. In tale contesto, l'utilizzo di un'architettura leggera consente di mantenere tempi di inferenza ridotti, rendendo il modello potenzialmente applicabile in scenari real-time. Il modello è stato inizializzato utilizzando pesi pre-addestrati (pretrained = True), sfruttando conoscenze apprese su dataset di grandi dimensioni. Questa strategia consente di migliorare la capacità di generalizzazione e di accelerare il processo di convergenza durante il *training* [14].

Dal punto di vista strutturale, il modello utilizza un input di dimensione 640×640 pixel, valore che rappresenta un compromesso tra dettaglio spaziale e costo computazionale. Inoltre, la configurazione del modello è stata adattata automaticamente dal framework Ultralytics in base al numero di classi presenti nel dataset, pari a 5 categorie di strumenti laparoscopici, permettendo una corretta definizione del layer di output della rete [7].

Il modello è stato addestrato utilizzando il task di tipo *detect*, con una configurazione standard del framework, che include l'utilizzo di tecniche di *data augmentation* e ottimizzazione automatica degli iperparametri. Non sono state apportate modifiche strutturali all'architettura di base, al fine di mantenere un'implementazione stabile e conforme alle best practice fornite dalla libreria. Questa configurazione ha permesso di ottenere un modello efficiente e robusto, in grado di adattarsi alle specificità del dataset laparoscopico e di garantire buone prestazioni in termini di rilevamento degli strumenti chirurgici.

3.6.3. Andamento del training e analisi della funzione di loss

Durante il processo di addestramento dei modelli è stato monitorato l'andamento delle principali componenti della funzione di *loss*, al fine di valutare la stabilità del training e la capacità di convergenza della rete neurale. In particolare, sono state analizzate la *box loss*, associata alla localizzazione delle *bounding boxes*, la *classification loss* (*cls loss),* relativa alla corretta assegnazione delle classi, e la *distribution focal loss* (*dfl loss*), utilizzata per migliorare la precisione nella regressione delle coordinate [7]. Come mostrato in Fig. 3.13, tutte le componenti della loss presentano una tendenza complessivamente decrescente nel corso delle epoche per entrambi i modelli YOLOv8 e YOLOv26, indicando una progressiva ottimizzazione dei parametri e una corretta fase di apprendimento.

La *box loss* quantifica l'errore nella localizzazione spaziale degli oggetti rilevati, misurando la discrepanza tra le bounding boxes predette dal modello e quelle di riferimento (*ground truth*), generalmente attraverso metriche basate sull'*Intersection over Union* (*IoU*).

Nel caso di YOLOv26, essa mostra una decrescita progressiva e regolare lungo tutto il training, passando da valori iniziali elevati (circa 1.3–1.4) fino a raggiungere un valore finale pari a 0.35496. Dopo una fase iniziale di rapida diminuzione, la curva entra in un regime di miglioramento più graduale, indicativo di un raffinamento progressivo della localizzazione degli oggetti.

Nel caso di YOLOv8, si osserva una dinamica simile ma con una convergenza più rapida verso valori inferiori, fino a circa 0,28. Tuttavia, nei grafici si evidenzia una maggiore distanza tra le curve di training e *validation*, suggerendo una minore stabilità del processo di apprendimento rispetto a YOLOv26. La maggiore coerenza tra *training* e *validation loss* osservata in YOLOv26 indica invece una migliore capacità di generalizzazione.

La *classification loss* misura l'errore associato all'assegnazione delle etichette di classe agli oggetti rilevati, confrontando la distribuzione di probabilità predetta con quella target.

Nel modello YOLOv26, il grafico evidenzia una dinamica particolarmente marcata: si osserva un rapido decremento nelle prime epoche (da valori superiori a 4 fino a circa 1 in poche iterazioni), seguito da una stabilizzazione su valori molto contenuti, fino al valore finale di 0,12444. Questo andamento riflette una rapida acquisizione della capacità discriminativa da parte del modello.

Nel caso di YOLOv8, la *classification loss* raggiunge valori ancora più bassi (circa 0.099), ma presenta una maggiore variabilità nelle fasi intermedie e una più evidente differenza tra *training* e *validation*, suggerendo una maggiore sensibilità alle fluttuazioni del dataset. Le leggere oscillazioni osservate in entrambi i modelli possono essere attribuite alla variabilità introdotta dalle tecniche di data augmentation, senza tuttavia compromettere la stabilità complessiva del training.

La *Distribution Focal Loss* rappresenta una funzione di perdita utilizzata per la regressione delle coordinate delle *bounding boxes* che modella la posizione dei bordi come una distribuzione discreta di probabilità.

Nel caso di YOLOv26, essa presenta un andamento monotonicamente decrescente e particolarmente regolare, raggiungendo un valore finale molto basso pari a 0.00366. Tale comportamento indica un miglioramento continuo e fine della regressione delle coordinate.

Nel caso di YOLOv8, la *dfl loss* raggiunge un valore finale leggermente inferiore (circa 0.0029), ma i grafici mostrano una maggiore distanza tra *training* e *validation* e una presenza più evidente di oscillazioni, suggerendo una minore stabilità del processo di ottimizzazione.

Modello	Box Loss	Cls Loss	DFL Loss
YOLOv8	0.28171	0.09981	0.00292
YOLOv26	0.35496	0.12444	0.00366

Figura 3.12 Valori finali delle principali componenti della funzione di loss e delle metriche di valutazione per i modelli YOLOv8 e YOLOv26

Nel complesso, il confronto tra i due modelli evidenzia due dinamiche di apprendimento differenti: YOLOv8 mostra una convergenza più rapida e valori di loss leggermente inferiori, mentre YOLOv26 presenta un andamento più regolare e stabile, con una migliore coerenza tra training e validation. Questo comportamento suggerisce che YOLOv26, pur non minimizzando la funzione di loss in modo più aggressivo, riesca a costruire rappresentazioni più robuste e generalizzabili, come confermato anche dalle prestazioni superiori in termini di accuratezza. Tali caratteristiche risultano particolarmente rilevanti nel contesto delle immagini laparoscopiche, in cui la variabilità delle condizioni visive richiede modelli in grado di mantenere elevata stabilità e affidabilità durante il processo di inferenza.

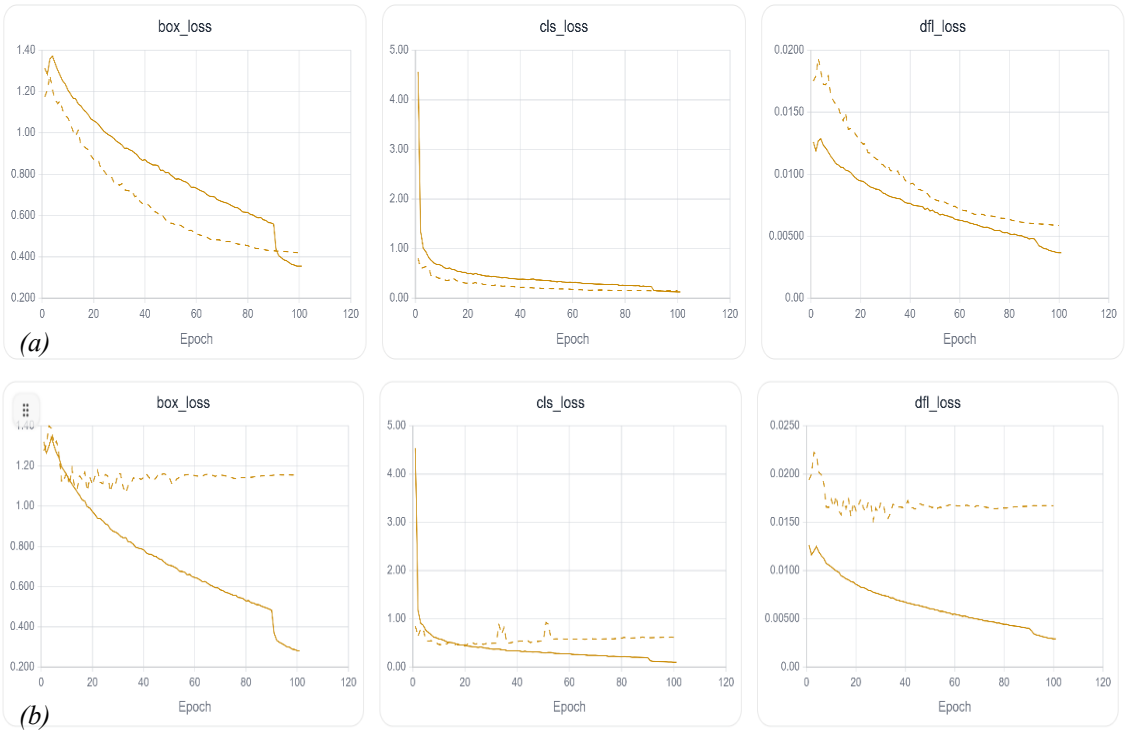


Figura 3.13 Andamento delle componenti della funzione di loss durante il training per i modelli YOLOv8 (a) e YOLOv26 (b).

3.6.4. Metriche di valutazione del modello

La valutazione delle prestazioni del modello rappresenta una fase cruciale per determinare l'efficacia del sistema di *object detection* sviluppato. Nel presente lavoro, le performance dei modelli sono state analizzate mediante l'utilizzo delle principali metriche comunemente adottate nel campo della computer vision, tra cui *precision*, *recall*, *Mean Average Precision (mAP*) e *Intersection over Union (IoU*) [25].

La *precision* misura la proporzione di predizioni corrette rispetto al totale delle predizioni effettuate dal modello ed è definita come:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

Dove TP rappresenta il valore appartenente ai veri positivi e FP quello dei falsi positivi. Un valore elevato di *precision* indica che il modello commette pochi errori di falsa identificazione.

Come mostrato in Fig. 5-13, nel caso di YOLOv26 la precision cresce rapidamente nelle prime epoche, passando da valori iniziali intorno a 0.75 fino a stabilizzarsi su valori molto elevati, prossimi a 0.99 nelle fasi finali del training. Questo andamento indica una progressiva riduzione dei falsi positivi e una crescente affidabilità del modello nelle predizioni.

Nel caso di YOLOv8, si osserva un incremento analogo nelle prime epoche, ma con una maggiore variabilità lungo il training e una stabilizzazione su valori inferiori, intorno a 0.89–0.90, evidenziando una minore stabilità complessiva.

La *recall* misura la capacità del modello di individuare tutti gli oggetti presenti nell'immagine:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

Dove TP rappresenta il valore appartenente ai veri positivi, FN quello dei falsi negativi.

Nel caso di YOLOv26, l'andamento mostra un incremento estremamente rapido nelle prime epoche, con valori che passano da circa 0,60 fino a stabilizzarsi anch'essi intorno a 0,99, indicando che il modello è in grado di rilevare quasi la totalità degli strumenti laparoscopici presenti nelle immagini.

YOLOv8 presenta un comportamento simile nelle fasi iniziali, ma con oscillazioni più evidenti e una convergenza verso valori inferiori, pari a circa 0,91–0,92, suggerendo una minore capacità di mantenere prestazioni stabili nel tempo.

La *mean Average Precision* (mAP) rappresenta una misura complessiva delle prestazioni del modello. In particolare, il valore di *mAP@0.5* considera corrette le predizioni con IoU superiore a 0.5, mentre *mAP@0.5:0.95* fornisce una valutazione più rigorosa mediando su diverse soglie di IoU.

Come evidenziato in Fig. 3.15, nel caso di YOLOv26 il valore di *mAP@0.5* cresce rapidamente nelle prime epoche fino a raggiungere valori prossimi a 0.99, evidenziando un'elevata capacità del modello di individuare correttamente gli oggetti. Il valore di *mAP@0.5:0.95*, più restrittivo, mostra una crescita più graduale ma costante, fino a stabilizzarsi intorno a 0.95, confermando una notevole accuratezza anche nella localizzazione precisa delle *bounding boxes*.

Nel caso di YOLOv8, il *mAP@0.5* raggiunge valori elevati ma inferiori (circa 0.94), mentre il *mAP@0.5:0.95* si stabilizza intorno a 0.75, indicando una minore precisione nella localizzazione a soglie più stringenti.

L'*Intersection over Union* (IoU) misura il grado di sovrapposizione tra *bounding box* predetta e reale, rappresentando un indicatore diretto della qualità della localizzazione. I valori elevati di *mAP@0.5:0.95* osservati nel caso di YOLOv26 implicano che il modello raggiunge livelli di IoU elevati anche per soglie più restrittive, a testimonianza della precisione spaziale raggiunta. Al contrario, i valori inferiori osservati per YOLOv8 suggeriscono una maggiore difficoltà nel mantenere un'elevata accuratezza geometrica nelle predizioni.

Modello	Precision	Recall	mAP 0.5	mAP 0.5:0.95
YOLOv8	0.8973	0.9136	0.9406	0.7525
YOLOv26	0.9892	0.9918	0.9944	0.9529

Figura 3.14 Valori finali delle principali metriche di valutazione per i modelli YOLOv8 e YOLOv26

Complessivamente, l'insieme delle metriche analizzate evidenzia due comportamenti distinti: YOLOv8 mostra una convergenza più rapida ma caratterizzata da maggiore variabilità, mentre YOLOv26 presenta un processo di apprendimento più stabile e progressivo, con una migliore coerenza tra le metriche e prestazioni complessive superiori. L'assenza di degradi nelle prestazioni e la stabilità delle curve suggeriscono un'elevata capacità di generalizzazione del modello YOLOv26, senza evidenti fenomeni di *overfitting*. I risultati ottenuti confermano quindi che YOLOv26 è altamente efficace nel riconoscimento e nella localizzazione degli strumenti laparoscopici, garantendo prestazioni elevate anche in scenari complessi.

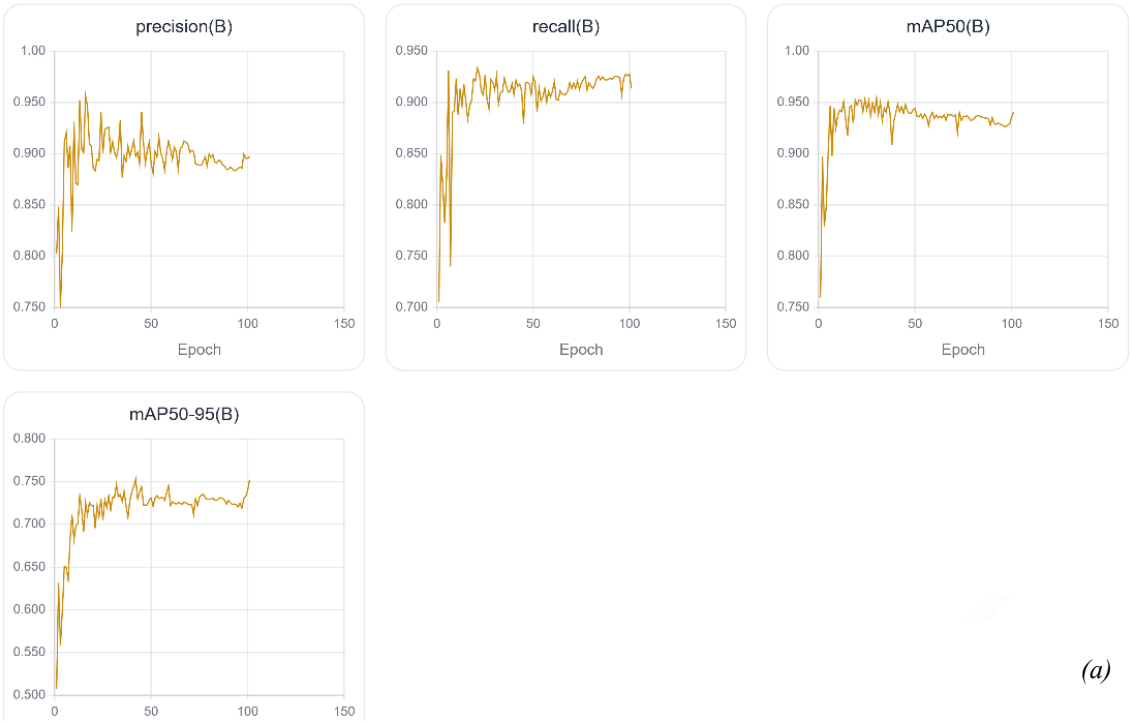


Figura 3.15 Andamento delle metriche di valutazione durante il training: (a) YOLOv8, (b) YOLOv26

La matrice di confusione per la rete YOLOv26 evidenzia inoltre una forte concentrazione dei valori lungo la diagonale principale, indicativa di una classificazione corretta per la quasi totalità dei campioni. In particolare, le classi *clipper*, *grasper*, *hook* e *irrigator* presentano percentuali di classificazione pari al 100%, mentre la classe *scissor* mostra un errore minimo (circa 1%), suggerendo una lieve ambiguità residua. L'assenza quasi totale di valori fuori diagonale indica che il modello non confonde significativamente le diverse classi, il che è particolarmente rilevante in contesti applicativi come quello laparoscopico, dove strumenti diversi possono presentare caratteristiche visive simili. Nel complesso, la matrice conferma una capacità discriminativa estremamente elevata, coerente con i risultati osservati nelle metriche globali.

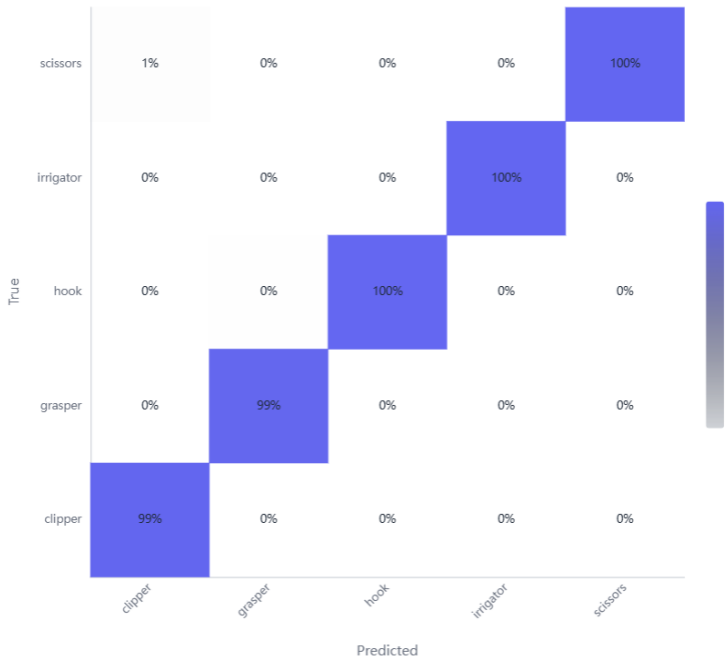


Figura 3.16 Matrice di confusione della rete YOLOv26

Alla luce dei risultati ottenuti, il modello YOLOv26 risulta complessivamente più adatto rispetto a YOLOv8 per il task di rilevamento e tracciamento di strumenti laparoscopici. Pur in presenza di differenze contenute nei valori delle funzioni di loss, le metriche di valutazione evidenziano un miglioramento significativo delle prestazioni complessive, in particolare per quanto riguarda il *mAP@0.5:0.95*, che rappresenta un indicatore più rigoroso della qualità della localizzazione. Tale aspetto risulta particolarmente rilevante

nel contesto laparoscopico, in cui la precisione nella stima delle *bounding boxes* è fondamentale per garantire un tracciamento affidabile degli strumenti nel tempo. La maggiore accuratezza ottenuta da YOLOv26 si traduce infatti in una migliore continuità del tracking e in una riduzione degli errori di localizzazione, soprattutto in presenza di occlusioni, variazioni di scala e condizioni visive complesse. Inoltre, il comportamento più stabile osservato durante il training suggerisce una maggiore robustezza del modello, che si riflette in prestazioni più consistenti anche su dati non visti. Questo elemento è particolarmente importante in applicazioni reali, dove la variabilità delle condizioni operative richiede modelli in grado di mantenere elevata affidabilità.

La scelta della rete YOLOv26 è stata inoltre supportata da un'analisi qualitativa fatta a seguito della loro applicazione sui video laparoscopici.

Nell'applicazione della rete YOLOv8 si riscontrano visivamente diverse criticità che possono influire negativamente sulle prestazioni del modello. In primo luogo, si osservano basse confidence score, indice di una limitata sicurezza nelle predizioni, probabilmente dovuta a scarsa qualità del training o a una distribuzione dei dati non adeguatamente rappresentativa. Inoltre, sono presenti bounding box sovrapposte e multiple sugli stessi oggetti, segnale di una possibile inefficacia della procedura di *Non-Maximum Suppression (NMS)* o di una difficoltà del modello nel distinguere istanze univoche. Un'ulteriore problematica è legata alla variabilità visiva dell'ambiente chirurgico, caratterizzato da riflessi, fluidi biologici e texture poco omogenee, che possono generare confusione nel modello e portare a falsi positivi o a una localizzazione imprecisa degli strumenti. Si nota anche una possibile ambiguità tra classi simili (ad esempio strumenti chirurgici con forme affini), che suggerisce una separabilità non ottimale nello spazio delle feature. Infine, la presenza di oggetti parzialmente occlusi o fuori fuoco evidenzia una limitata robustezza del modello rispetto a condizioni realistiche di utilizzo, indicando la necessità di dataset più vari e tecniche di data augmentation più avanzate.

La rete YOLOv26 si è dimostrata significativamente migliore rispetto alla precedente. In particolare, le predizioni risultano caratterizzate da confidence score elevate, indicando una maggiore sicurezza del modello nel riconoscimento degli strumenti chirurgici. Le *bounding box* appaiono ben localizzate e non si riscontrano sovrapposizioni multiple sullo

stesso oggetto, suggerendo un corretto funzionamento della procedura di *Non-Maximum Suppression* e una buona separazione tra le istanze rilevate. Il modello gestisce meglio la distinzione tra classi rispetto a YOLOv8, e risulta molto meno sensibile a fattori come illuminazione, occlusioni parziali e variabilità anatomica.

Nel complesso, YOLOv26 offre quindi un miglior compromesso tra accuratezza, stabilità e capacità di generalizzazione, rendendolo la scelta più appropriata per applicazioni di *computer-assisted surgery* basate sul rilevamento e tracciamento automatico degli strumenti laparoscopici.

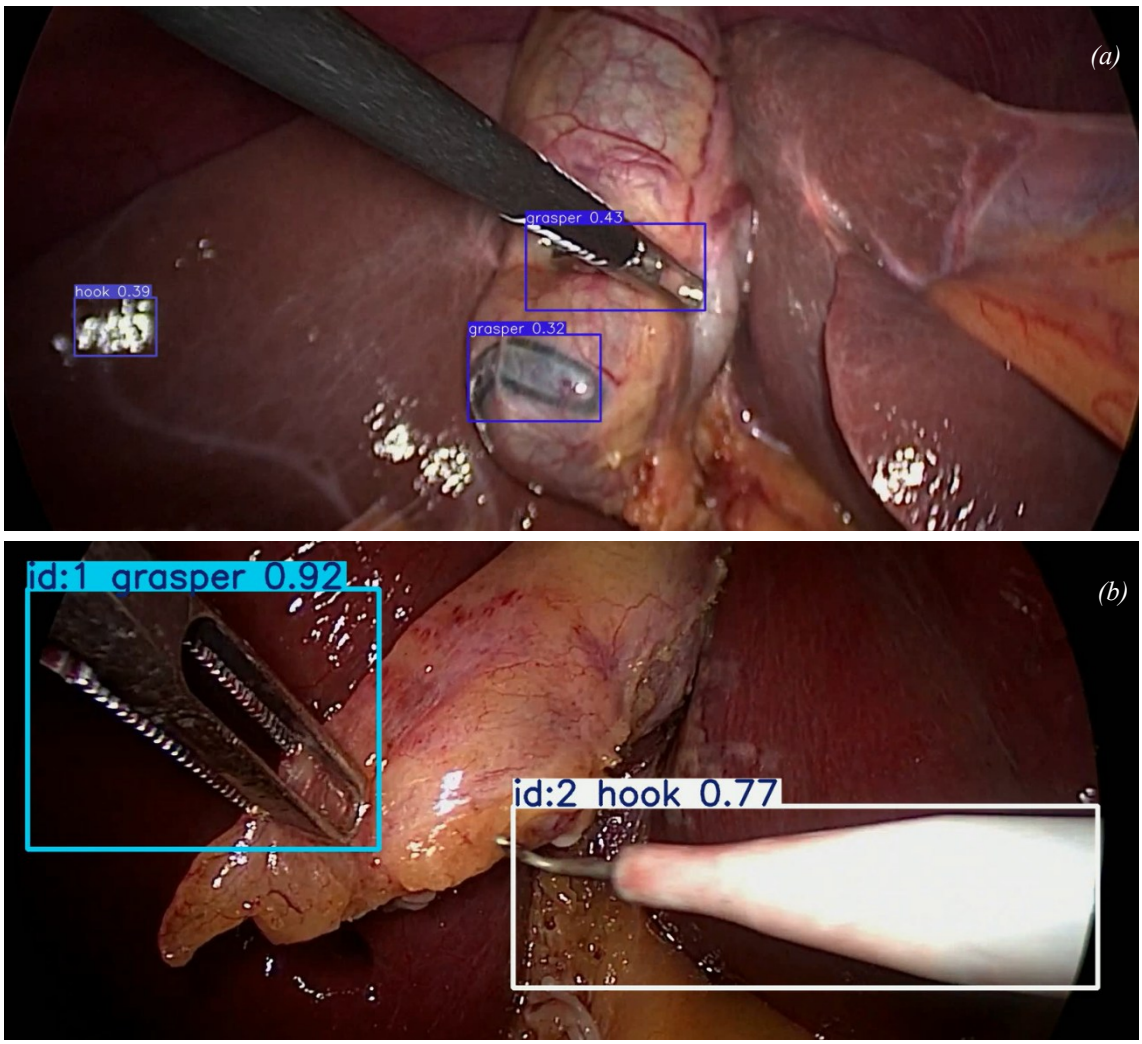


Figura 3.17 Confronto qualitativo tra le prestazioni della reteYOLOv8 (a) e YOLOv26 (b)

4. Multi-Object Tracking (MOT)

Nel contesto della visione artificiale applicata all'analisi video, il rilevamento degli oggetti costituisce una fase fondamentale ma non sufficiente per la comprensione dinamica della scena. In particolare, algoritmi di *object detection* come YOLOv26 operano a livello di singolo *frame*, producendo per ciascuna immagine un insieme di *bounding box* e relative classi, senza tuttavia fornire informazioni sulla continuità temporale delle istanze rilevate. Tale limitazione rende necessario l'impiego di tecniche di *multi-object tracking* (MOT), il cui obiettivo è associare le *detection* tra *frame* consecutivi, assegnando a ciascun oggetto un identificatore persistente nel tempo [1].

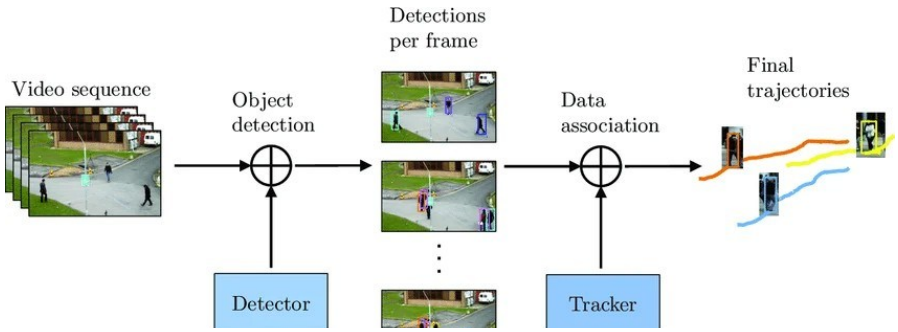


Figura 4.1 Schema del funzionamento del paradigma tracking-by-detection

I moderni sistemi MOT adottano prevalentemente un approccio *tracking-by-detection*, in cui un detector fornisce le osservazioni a ogni istante temporale, mentre un algoritmo di *tracking* si occupa di modellare il moto degli oggetti e di risolvere il problema di associazione tra osservazioni e tracce esistenti. Formalmente, dato un insieme di *detection* Dt al tempo t, il problema consiste nel determinare una funzione di assegnazione che minimizzi il costo complessivo tra osservazioni e tracce, garantendo al contempo la coerenza temporale delle identità. Le principali sfide del MOT includono la gestione delle occlusioni, le variazioni di illuminazione, il movimento della camera e la presenza di falsi positivi o falsi negativi nelle *detection*. Per affrontare tali problematiche, gli algoritmi moderni combinano modelli di predizione del moto, tipicamente basati su filtri di Kalman, con strategie avanzate di data *association* e, in alcuni casi, con l'utilizzo di feature di apparenza estratte tramite reti neurali profonde. Nel presente lavoro, il problema del *tracking* multi-oggetto viene affrontato integrando YOLOv26 con due

algoritmi rappresentativi dello stato dell'arte recente, ovvero BoT-SORT e ByteTrack. Tali metodi implementano strategie differenti per l'associazione delle *detection* e la gestione delle tracce, offrendo un interessante confronto tra approcci basati su feature di apparenza e approcci basati esclusivamente su informazioni geometriche e di confidenza. Nel framework YOLOv26, il modello di *object detection* è responsabile esclusivamente dell'identificazione e della localizzazione degli oggetti all'interno di singoli *frame*, producendo in *output* un insieme di *detection* costituite da *bounding box*, classi e punteggi di confidenza. Tuttavia, YOLOv26 non implementa nativamente un meccanismo di *tracking* multi-oggetto, rendendo necessario l'utilizzo di algoritmi esterni per garantire la continuità temporale delle istanze rilevate. Nei moderni *framework* di visione artificiale, come quelli sviluppati da Ultralytics, il *tracking* viene integrato secondo un'architettura modulare, in cui il *detector* e il *tracker* operano come componenti separati ma interconnessi. In particolare, YOLOv26 fornisce in input al modulo di *tracking* le *detection* generate per ciascun frame, mentre algoritmi come BoT-SORT e ByteTrack si occupano di associare tali osservazioni nel tempo, costruendo tracce coerenti. È importante sottolineare che, nelle implementazioni più recenti, YOLOv26 è tipicamente configurato per lavorare con tracker avanzati come BoT-SORT e ByteTrack, che rappresentano lo stato dell'arte nel multi-object tracking real-time. Questi algoritmi vengono selezionati in base alle esigenze applicative: BoT-SORT risulta particolarmente efficace in scenari complessi grazie all'utilizzo di feature di apparenza, mentre ByteTrack offre un'elevata efficienza computazionale e una buona robustezza sfruttando anche le *detection* a bassa confidenza. Dal punto di vista operativo, l'intero sistema può essere descritto come una pipeline sequenziale:

$$\text{Frame}_t \overset{\text{YOLOv26}}{\rightarrow} D_t \overset{\text{Tracker}}{\rightarrow} T_t$$

Dove:

- D_t rappresenta l'insieme delle *detection* al tempo t
- T_t l'insieme delle tracce aggiornate dal *tracker*.

Questa separazione tra *detection* e *tracking* consente una maggiore flessibilità del sistema, permettendo di sostituire o aggiornare il modulo di tracking senza modificare il detector, e viceversa.

4.1. BoT-SORT

BoT-SORT (*Bag of Tricks SORT*) rappresenta un'evoluzione avanzata degli algoritmi di *tracking-by-detection*, progettata per migliorare le prestazioni dei metodi basati su SORT attraverso l'integrazione di tecniche di *re-identification* (ReID) e una gestione più accurata del moto della scena [30]. In una pipeline basata su YOLOv26, BoT-SORT riceve in input le *detection* Dt generate dal detector e aggiorna iterativamente l'insieme delle tracce Tt, secondo un approccio online. L'obiettivo del *tracking* è quello di mantenere nel tempo l'identità degli oggetti rilevati, costruendo delle tracce, cioè sequenze temporali di *bounding box* che rappresentano lo stesso oggetto nei frame successivi. Per fare ciò, BoT-SORT mantiene un insieme di tracce attive e, per ogni nuovo frame, deve associare le nuove *detection* alle tracce esistenti, risolvendo un problema di corrispondenza tra osservazioni e modelli predetti. Per gestire questa associazione, l'algoritmo utilizza un filtro di Kalman, ovvero un modello matematico probabilistico che consente di stimare lo stato futuro di un sistema dinamico (in questo caso la posizione e la dimensione dell'oggetto) a partire dalle osservazioni passate. In particolare, il filtro predice la posizione attesa di ciascun oggetto nel frame successivo, permettendo di limitare la ricerca delle possibili associazioni. Un elemento distintivo di BoT-SORT è l'introduzione della *Global Motion Compensation* (GMC), ovvero una tecnica che stima il movimento globale della scena tra due *frame* consecutivi, tipicamente causato dallo spostamento della camera. Questa stima viene utilizzata per correggere le posizioni predette degli oggetti, evitando che il movimento della camera venga erroneamente interpretato come movimento degli oggetti stessi. L'associazione tra tracce e *detection* avviene combinando due tipi di informazione. Da un lato, viene utilizzata una misura geometrica chiamata *Intersection over Union* (IoU), che quantifica il grado di sovrapposizione tra due *bounding box*. Dall'altro lato, vengono impiegate informazioni di apparenza, ottenute tramite feature di *re-identification* (ReID), cioè rappresentazioni numeriche estratte da una rete neurale che descrivono l'aspetto visivo dell'oggetto (ad esempio colore e forma). Queste feature consentono di riconoscere lo stesso oggetto anche in presenza di variazioni di posizione o di parziali occlusioni. Le informazioni geometriche e di apparenza vengono combinate per costruire una matrice di costo che rappresenta quanto ogni *detection* sia compatibile con ciascuna traccia. Il problema di

associazione viene quindi risolto tramite l'algoritmo, un metodo di ottimizzazione che individua la corrispondenza ottimale tra due insiemi, minimizzando il costo complessivo

Figura 4.2 Schema del funzionamento di BoT-SORT: le detection vengono associate alle tracce tramite una combinazione di predizione del moto (Kalman filter), informazioni geometriche (IoU) e feature di apparenza (ReID).

[1]. Infine, BoT-SORT introduce una serie di miglioramenti pratici, tra cui l'aggiornamento progressivo delle feature di apparenza e l'utilizzo di soglie adattive per filtrare associazioni non affidabili. Questi accorgimenti consentono di ridurre i cosiddetti *ID switch* (cambiamenti errati di identità) e di migliorare la continuità delle tracce nel tempo, rendendo l'algoritmo particolarmente efficace in scenari complessi caratterizzati da elevata densità di oggetti e frequenti occlusioni [30].

4.2. ByteTrack

ByteTrack è un algoritmo di *multi-object tracking* appartenente alla categoria *tracking-by-detection*, che introduce una strategia innovativa per l'associazione delle *detection* basata sull'utilizzo di tutte le osservazioni prodotte dal detector, incluse quelle a bassa confidenza [31]. In una pipeline basata su YOLOv26, il detector fornisce per ogni frame un insieme di *detection*, ossia bounding box (rettangoli che delimitano gli oggetti nell'immagine) accompagnati da un punteggio di confidenza che indica quanto il modello è sicuro della presenza dell'oggetto. A differenza dei metodi tradizionali, che tendono a scartare le *detection* con bassa confidenza per ridurre i falsi positivi, ByteTrack sfrutta anche queste informazioni, osservando che esse possono comunque contenere oggetti reali temporaneamente difficili da rilevare, ad esempio a causa di occlusioni, rumore o variazioni di illuminazione. Per questo motivo, le *detection* vengono suddivise in due insiemi: D^{high}, contenente le *detection* con confidenza elevata, e D^{low}, contenente quelle con confidenza più bassa. Il processo di *tracking* si basa sulla costruzione e sull'aggiornamento di tracce*,* ovvero sequenze temporali che rappresentano lo stesso oggetto nei diversi frame. Anche in ByteTrack, come in altri metodi MOT, viene utilizzato un filtro di Kalman, cioè un modello probabilistico che consente di predire la posizione futura di ciascun oggetto sulla base delle osservazioni precedenti. Questa predizione permette di limitare lo spazio delle possibili associazioni tra tracce e nuove *detection*.

L'elemento distintivo di ByteTrack è il processo di *data association* (associazione dei dati), che avviene in due fasi successive. Nella prima fase, le tracce esistenti vengono associate alle *detection* ad alta confidenza D^{high}, utilizzando una metrica basata su *Intersection over Union* (IoU), che misura la sovrapposizione tra *bounding box.* L'associazione ottimale viene determinata tramite l'algoritmo ungherese, che consente di minimizzare il costo complessivo di assegnazione tra tracce e *detection*. Nella seconda fase, le tracce che non sono state associate nella fase precedente vengono confrontate con le *detection* a bassa confidenza D^{low}. Questo passaggio consente di recuperare oggetti che non sono stati rilevati con alta sicurezza, riducendo il rischio di perdita della traccia (*track fragmentation*) e migliorando la continuità temporale delle identità. In questo modo, ByteTrack riesce a mantenere tracce più stabili anche in condizioni difficili, senza

introdurre un numero significativo di falsi positivi [31]. Formalmente, il problema di associazione può essere modellato come un problema di ottimizzazione combinatoria:

$$\min_{\pi} \sum_{i,j} c_{ij} \cdot \pi_{ij} \tag{8}$$

dove c_{ij}rappresenta il costo di associazione tra la traccia ie la detection j, tipicamente derivato dalla sovrapposizione IoU, e π_{ij}indica se l'associazione viene effettuata o meno. A differenza di algoritmi come BoT-SORT, ByteTrack non utilizza feature di apparenza estratte tramite reti neurali profonde (ReID), ma si basa esclusivamente su informazioni geometriche e sui punteggi di confidenza delle *detection*. Questa scelta rende l'algoritmo più semplice dal punto di vista computazionale e particolarmente adatto a contesti *real-time*, pur mantenendo prestazioni elevate in termini di accuratezza e stabilità delle tracce.

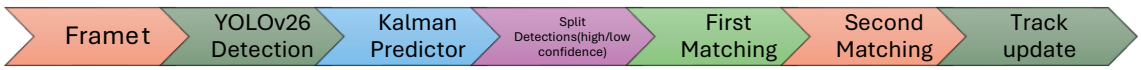


Figura 4.3 Pipeline dell'algoritmo ByteTrack: le detection generate dal detector vengono suddivise in alta e bassa confidenza e associate alle tracce in due fasi successive, utilizzando una predizione del moto basata su filtro di Kalman e un processo di assegnazione ottimale

4.3. Scelta dell'algoritmo di tracking

Nel presente lavoro, il problema del *tracking* multi-oggetto è stato affrontato integrando il modello di *object detection* YOLOv26 con un algoritmo appartenente alla categoria *tracking-by-detection*, approccio che rappresenta lo standard nei sistemi moderni di *multi-object* tracking in quanto consente di separare il problema della rilevazione degli oggetti da quello della loro associazione temporale [1].

In particolare, è stato adottato l'algoritmo BoT-SORT (*Bag of Tricks SORT*), selezionato per le sue prestazioni avanzate nello stato dell'arte del tracking in tempo reale e per la sua

integrazione nativa nel *framework* Ultralytics, dove è utilizzato come soluzione predefinita nella modalità *track* [30]. La scelta di BoT-SORT è stata ulteriormente motivata dalle caratteristiche specifiche del contesto applicativo considerato. L'analisi di video laparoscopici presenta infatti condizioni visive complesse, tra cui frequenti occlusioni, variazioni di illuminazione, riflessi speculari e movimenti non lineari degli strumenti chirurgici. In tali scenari, approcci basati esclusivamente su informazioni geometriche, come la sovrapposizione tra bounding box, possono risultare insufficienti per garantire la corretta continuità delle tracce nel tempo. Rispetto ad algoritmi come ByteTrack (5), che si basano prevalentemente su informazioni geometriche e sui punteggi di confidenza delle *detection*, BoT-SORT introduce l'utilizzo di feature di apparenza tramite modelli di *re-identification* (ReID). Questo consente di mantenere l'identità degli oggetti anche in presenza di occlusioni temporanee o variazioni significative della posizione, condizioni particolarmente frequenti durante le procedure chirurgiche. Inoltre, BoT-SORT integra un meccanismo di *Global Motion Compensation*[7] (GMC), che permette di stimare e compensare il movimento globale della scena causato dallo spostamento della telecamera laparoscopica. Questo aspetto risulta fondamentale in ambito chirurgico, dove il movimento della camera è continuo e può generare variazioni apparenti nella posizione degli strumenti, rendendo più complesso il problema di associazione. Un ulteriore vantaggio di BoT-SORT risiede nella combinazione di informazioni geometriche e di apparenza nella costruzione della matrice di costo per la *data association*, consentendo una maggiore robustezza rispetto a metodi basati esclusivamente su IoU. ByteTrack, pur offrendo un'elevata efficienza computazionale e buone prestazioni in scenari standard, può risultare meno robusto in presenza di occlusioni prolungate o in contesti caratterizzati da interazioni frequenti tra oggetti.

Alla luce di queste considerazioni, BoT-SORT rappresenta una scelta adeguata per il contesto laparoscopico, in quanto consente di ottenere una maggiore stabilità delle tracce e una gestione più efficace delle criticità tipiche dell'ambiente chirurgico, permettendo di estendere il sistema dalla semplice rilevazione degli oggetti all'analisi dinamica della scena.

[7] Tecnica che stima e corregge il movimento globale della scena tra frame consecutivi, al fine di separarlo dal movimento reale degli oggetti e migliorare l'accuratezza del tracking.

5. Analisi qualitativa del tracking multi-oggetto

L'integrazione del modello YOLOv26 con un algoritmo di *tracking* multi-oggetto ha consentito di estendere il sistema dall'analisi statica *frame-by-frame* a una rappresentazione dinamica della scena. In particolare, seguendo il paradigma *tracking-by-detection*, il modello di *object detection* fornisce, per ciascun frame, un insieme di *bounding box* associate alle classi degli strumenti e ai relativi punteggi di confidenza, mentre il modulo di *tracking* si occupa dell'associazione temporale delle *detection* e dell'assegnazione di identificatori persistenti agli oggetti. L'output del sistema consiste quindi in una rappresentazione visiva della scena, nella quale ogni oggetto rilevato viene non solo localizzato, ma anche seguito nel tempo attraverso l'assegnazione di ID univoci. Questo consente di mantenere la continuità delle istanze tra frame consecutivi e di distinguere efficacemente tra strumenti diversi presenti simultaneamente nel campo visivo.

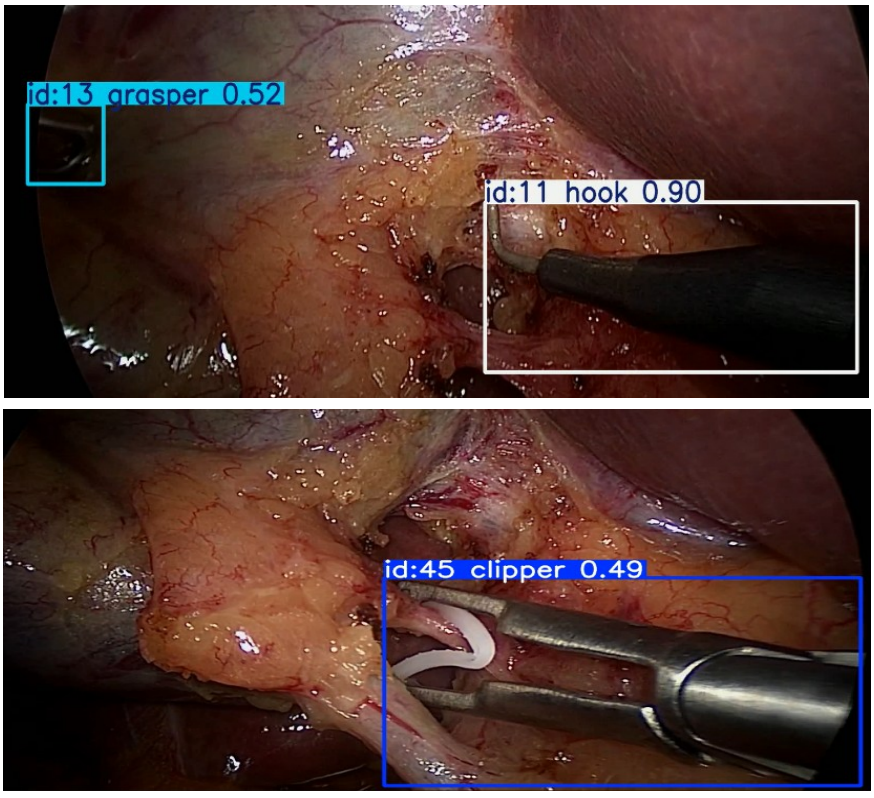


Figura 5.1 Esempio di tracking multi-oggetto in una sequenza laparoscopica: il sistema rileva e traccia simultaneamente strumenti chirurgici, assegnando identificatori persistenti (ID) e punteggi di confidenza a ciascuna detection.

Oltre alla visualizzazione grafica dei risultati, il sistema produce anche file di output in formato testuale, nei quali ciascuna *detection* è rappresentata mediante una riga contenente l'identificativo dell'oggetto, la classe e le coordinate normalizzate della bounding box. In particolare, ogni riga del file `.txt` può essere espressa nel formato:

$$class\ x_{center}\ y_{center}\ width\ height\ ID$$

dove *ID* rappresenta l'identificatore assegnato dal tracker, mentre le coordinate sono espresse in forma normalizzata rispetto alle dimensioni dell'immagine e assumono valori compresi tra 0 e 1. Questo formato consente una rappresentazione compatta e strutturata delle *detection*, rendendo possibile l'analisi temporale degli oggetti anche in assenza della componente visiva.

Dall'analisi qualitativa dei risultati emerge che il sistema è in grado di garantire una buona stabilità delle tracce, mantenendo gli identificatori coerenti anche in presenza di movimenti della camera e variazioni di prospettiva. In particolare, l'utilizzo di un algoritmo avanzato come BoT-SORT, basato su modelli di predizione del moto e feature di apparenza, consente di migliorare la continuità delle tracce rispetto ai metodi tradizionali. Il sistema dimostra inoltre una buona capacità di gestione delle occlusioni temporanee, riuscendo in molti casi a riassociare correttamente le *detection* allo stesso oggetto una volta che questo torna visibile. Questo aspetto risulta particolarmente rilevante nel contesto laparoscopico, dove gli strumenti interagiscono frequentemente tra loro e con i tessuti, generando condizioni di visibilità parziale. Tuttavia, in condizioni particolarmente complesse, caratterizzate da occlusioni prolungate o movimenti rapidi degli strumenti, possono verificarsi occasionali perdite di traccia o cambiamenti di identificatore (*ID switch*). Tali fenomeni risultano comunque limitati e non compromettono in modo significativo la qualità complessiva del sistema. Nel complesso, i risultati evidenziano come l'integrazione tra YOLOv26 e un algoritmo di *tracking* multi-oggetto consenta di ottenere un sistema efficace per il monitoraggio dinamico degli strumenti chirurgici, offrendo una rappresentazione coerente e continua delle istanze nel tempo anche in condizioni visive complesse.

6. Post-processing dei dati e calcolo delle metriche

L'algoritmo sviluppato in questo lavoro ha l'obiettivo di estrarre informazioni cinematiche a partire dal *tracking* bidimensionale degli strumenti chirurgici, ottenuto tramite la rete YOLOv26. In particolare, lo script MATLAB implementa una pipeline completa che, a partire dai risultati di *detection* e *tracking* organizzati in file testuali associati ai singoli *frame*, produce come *output* un insieme di indicatori quantitativi esportati in formato Excel.

Dal punto di vista applicativo, l'algoritmo consente di trasformare una rappresentazione puramente visiva del movimento degli strumenti in una descrizione numerica delle loro traiettorie, utile per analizzare il comportamento dinamico durante l'intervento. Per questo motivo, la pipeline è strutturata in modo modulare e comprende le seguenti fasi: definizione dei parametri, lettura sincronizzata di video e annotazioni, ricostruzione delle tracce, stima delle grandezze cinematiche, aggregazione per classe e per lato del frame e, infine, salvataggio dei risultati.

Il comportamento dell'algoritmo è regolato da una serie di parametri specifici, che svolgono un ruolo fondamentale sia nella selezione delle tracce utilizzabili sia nel livello di filtraggio applicato ai dati. Tra i principali parametri si considerano:

- minVisibleTime_s = 2.0, che impone una durata minima di visibilità dell'ID pari a 2 secondi. Le *detection* fornite da YOLO possono risultare instabili o sporadiche, ad esempio a causa di una comparsa parziale dello strumento nel campo visivo o di errori temporanei del modello. L'introduzione di una durata minima consente quindi di selezionare solo sequenze sufficientemente lunghe da rappresentare un movimento reale;
- minPath_px = 10, che permette di filtrare traiettorie in cui lo spostamento osservato è confrontabile con il rumore intrinseco della localizzazione YOLO. In assenza di questo vincolo, piccole oscillazioni nelle coordinate, dovute a jitter[8] del *bounding box*, potrebbero essere erroneamente interpretate come movimento significativo;

[8] variazione casuale e non sistematica della posizione stimata di un oggetto tra campioni consecutivi, dovuta a rumore o imprecisioni del sistema di rilevamento.

- smoothWindow = 7, che definisce la dimensione della finestra utilizzata per lo *smoothing* delle coordinate. La rete non garantisce una perfetta continuità nella posizione stimata tra frame consecutivi, soprattutto in presenza di variazioni di illuminazione, deformazioni dello strumento o cambiamenti di prospettiva. L'applicazione di una media mobile consente quindi di attenuare tali fluttuazioni, producendo una traiettoria più regolare e più adatta al calcolo delle derivate;
- maxGapFramesForKinematics = 1, che stabilisce che il calcolo della cinematica venga effettuato esclusivamente tra *frame* consecutivi, evitando intervalli temporali più ampi. In presenza di mancata *detection*, si crea infatti un intervallo temporale non osservato che impedisce di descrivere correttamente il moto dello strumento; il parametro consente quindi di escludere automaticamente tali casi, evitando che vengano utilizzati intervalli discontinui che porterebbero a stime non rappresentative.
- minSamplesSpeedPerTrack, minSamplesAccelPerTrack e minSamplesJerkPerTrack, che definiscono il numero minimo di campioni necessari per considerare affidabile il calcolo, rispettivamente, di velocità, accelerazione e jerk. Sequenze troppo brevi tendono infatti a produrre stime numericamente instabili.

Per migliorare ulteriormente l'affidabilità delle grandezze cinematiche, viene applicata una procedura di rimozione degli *outlier* basata sull'intervallo interquartile (IQR)[9]. I dati vengono ordinati e suddivisi tramite i quartili: il primo quartile $(Q1)$ rappresenta il valore al di sotto del quale si trova il 25% dei dati, mentre il terzo quartile $(Q3)$ rappresenta il valore al di sotto del quale si trova il 75%. L'intervallo interquartile è definito come:

$$IQR = Q3 - Q1 \tag{9}$$

Un valore viene considerato anomalo se risulta esterno all'intervallo:

$$[Q1 - k \cdot IQR,\ Q3 + k \cdot IQR] \tag{10}$$

[9] misura statistica utilizzata per descrivere la dispersione dei dati centrali.

con $k = 2.5$; tale scelta è coerente con applicazioni in cui i dati presentano variabilità intrinseca elevata, per le quali è preferibile adottare criteri di filtraggio meno aggressivi. Questo criterio consente di eliminare valori non rappresentativi, migliorando la robustezza delle stime.

Dal video vengono estratte alcune informazioni fondamentali, tra cui il frame rate (fps), la larghezza (W) e l'altezza (H) del frame. Le coordinate normalizzate fornite dal modello vengono convertite nel dominio immagine secondo:

$$x = x_{\text{norm}} \cdot W, y = y_{\text{norm}} \cdot H \tag{11}$$

Il numero totale di frame del video può essere stimato a partire dalla sua durata temporale e dal *frame rate*. In particolare, conoscendo la durata del video espressa in secondi e il numero di frame acquisiti ogni secondo, è possibile ottenere una stima del numero complessivo di frame come prodotto tra queste due grandezze.

Le traiettorie vengono memorizzate in una struttura contenente per ogni oggetto l'identificativo, la classe, i frame di osservazione, le coordinate del centro e la confidenza. L'associazione tra identificativo e traiettoria è gestita tramite una struttura di tipo dizionario (containers.Map), che consente un aggiornamento efficiente delle tracce lungo il tempo.

Terminata la fase di ricostruzione, il codice procede con l'analisi delle traiettorie. Per ciascuna traccia viene innanzitutto calcolata la durata di visibilità:

$$T_{\text{vis}} = \frac{N_f}{fps} \tag{11}$$

Dove T_{vis} rappresenta il tempo di visibilità della traccia, N_f il numero di frame in cui l'oggetto è osservato e fps il *frame rate* del video, espresso in frame al secondo.

Successivamente, se il numero di campioni è sufficiente, viene applicato uno *smoothing* delle coordinate. A partire dalle coordinate filtrate vengono calcolati gli spostamenti tra campioni consecutivi e la distanza euclidea:

$$d_k = \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \tag{12}$$

La lunghezza complessiva del percorso è quindi:

$$L = \sum_{k} d_k \tag{13}$$

La velocità viene calcolata come:

$$v_k = \frac{d_k}{\Delta t_k} \tag{14}$$

L'accelerazione è definita come variazione della velocità nel tempo:

$$a_k = \frac{v_{k+1} - v_k}{\bar{\Delta t}} \tag{15}$$

mentre il jerk è definito come variazione dell'accelerazione:

$$j_k = \frac{a_{k+1} - a_k}{\bar{\Delta t}} \tag{16}$$

Non tutte le traiettorie vengono utilizzate per l'analisi finale: vengono infatti considerate solo quelle che soddisfano criteri di validità cinematica, relativi alla durata, allo spostamento e al numero di campioni disponibili.

I risultati vengono quindi aggregati per classe e lato del frame. Il tempo totale di presenza viene calcolato considerando i frame unici:

$$T_{\text{tot}} = \frac{N_{\text{frame unici}}}{fps} \tag{17}$$

dove il numeratore rappresenta il numero di frame distinti in cui compare almeno una traccia della classe. La la percentuale di visibilità è definita come:

$$V = 100 \cdot \frac{N_{\text{frame unici}}}{N_{\text{frame processati}}} \tag{18}$$

Le grandezze cinematiche vengono sintetizzate attraverso valori medi e massimi; nel caso di accelerazione e jerk si considera il valore assoluto, in quanto l'interesse è rivolto all'intensità delle variazioni dinamiche.

Infine, i risultati vengono organizzati in una tabella ed esportati in formato Excel, rendendo i dati facilmente consultabili.

Lo script include inoltre diverse funzioni ausiliarie che migliorano la robustezza dell'implementazione, tra cui la lettura sicura dei file, la rimozione degli outlier tramite IQR, il calcolo protetto delle statistiche e l'ordinamento corretto dei file.

Dal punto di vista implementativo, l'algoritmo realizza una pipeline completa e coerente per l'analisi del movimento a partire da dati di tracking bidimensionali. La separazione tra ricostruzione delle tracce, stima delle grandezze cinematiche e aggregazione finale rende il codice facilmente interpretabile e modificabile.

È importante sottolineare che tutte le metriche sono espresse nel dominio immagine, ovvero in pixel e loro derivate rispetto al tempo. Sebbene ciò consenta confronti relativi tra strumenti, lati del frame o differenti video, tali misure non rappresentano ancora grandezze fisiche assolute. Un possibile sviluppo futuro consiste nella calibrazione della scena laparoscopica, al fine di convertire le coordinate in unità metriche reali.

7. Discussione dei risultati

Le metriche ricavate da quattro video di colecistectomia forniscono una prima panoramica di quelle che potrebbero essere le performance che permettono di discriminare le prestazioni tra esperti e novizi chirurghi.

7.1. Fase di esposizione

L'analisi della fase di esposizione della colecisti, condotta attraverso i parametri cinematici delle due pinze laparoscopiche, evidenzia una chiara differenziazione funzionale tra gli strumenti, interpretabile alla luce sia dei dati quantitativi sia dell'osservazione qualitativa del gesto chirurgico.

In particolare, la pinza sinistra mostra una maggiore variabilità nei parametri analizzati, con valori generalmente più elevati di percorso totale (TotalPath) e una maggiore dispersione nei valori di tempo totale, velocità media e jerk tra i diversi casi. Tale comportamento suggerisce un ruolo più dinamico dello strumento, caratterizzato da continui aggiustamenti della traiettoria e da una più intensa interazione con i tessuti.

Al contrario, la pinza destra presenta un comportamento più stabile e ripetibile, con valori di tempo, velocità e percorso che risultano tra loro più omogenei nei diversi casi analizzati. Questa minore variabilità suggerisce un utilizzo più controllato e meno esplorativo dello strumento, compatibile con una funzione di stabilizzazione e mantenimento della tensione sui tessuti durante la fase di esposizione.

In particolare, il caso 2 si distingue per valori più contenuti di tempo totale e percorso, associati a valori relativamente elevati di velocità media, accelerazione e jerk. Questo suggerisce una strategia più rapida e dinamica, caratterizzata da movimenti più intensi e meno regolari, in cui la riduzione del tempo di esecuzione è ottenuta attraverso un aumento della variabilità del gesto. Al contrario, i casi 1 e 4 mostrano tempi più elevati e valori di percorso maggiori, accompagnati da una dinamica più contenuta, indicando un approccio più graduale e controllato, con movimenti più continui e meno bruschi.

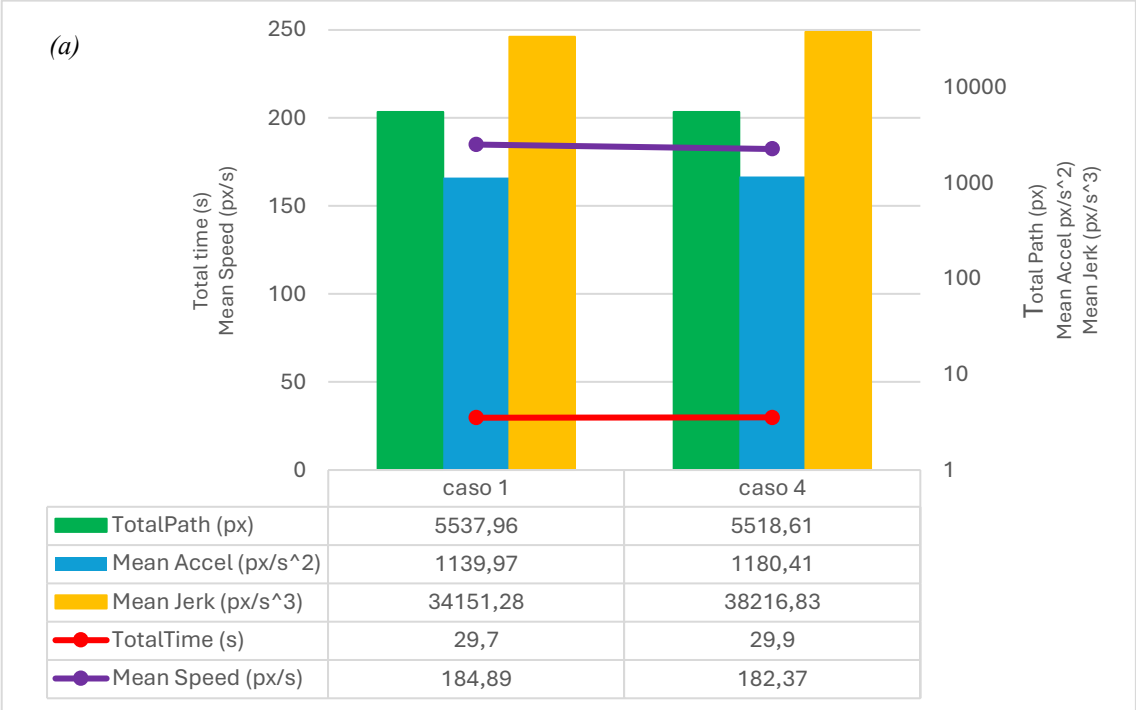
Il caso 0 presenta valori intermedi o inferiori sia in termini di tempo sia di parametri dinamici, configurandosi come una modalità operativa più semplice o più diretta, caratterizzata da un numero ridotto di aggiustamenti e da una minore intensità del movimento. Nel complesso, si osserva come nella fase di esposizione il principale fattore discriminante tra i casi sia rappresentato dal compromesso tra rapidità e fluidità del gesto: configurazioni più veloci sono associate a valori più elevati di accelerazione e jerk, mentre approcci più lenti risultano caratterizzati da una maggiore regolarità del movimento, coerente con la necessità di mantenere una trazione stabile sul tessuto.

Tuttavia, l'interpretazione del ruolo funzionale delle due pinze non può basarsi esclusivamente sui parametri cinematici. L'osservazione dei video evidenzia infatti come la pinza sinistra sia costantemente presente nel campo operatorio, mentre la pinza destra risulta talvolta meno visibile o meno coinvolta in modo continuo. Questo elemento suggerisce che la pinza sinistra possa svolgere un ruolo di sostegno ed esposizione del tessuto, mantenendo una trazione costante necessaria alla corretta visualizzazione del piano anatomico. Tale ipotesi risulta coerente con la pratica chirurgica laparoscopica, in cui la mano non dominante è frequentemente deputata alla stabilizzazione del campo operatorio, mentre la mano dominante (che supponiamo essere la destra in tutti i video) è destinata nelle fasi successive all'utilizzo di strumenti operativi più specifici, quali ad esempio l'uncino per la dissezione.

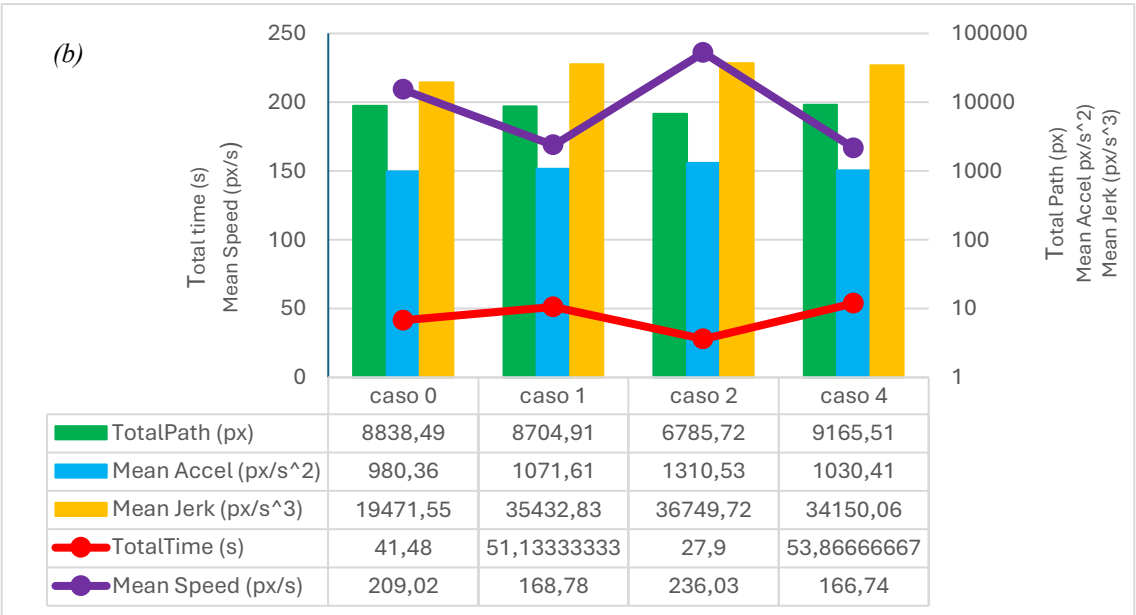
Alla luce di queste considerazioni, emerge un apparente paradosso interpretativo: se da un lato la pinza sinistra presenta parametri cinematici compatibili con un ruolo più attivo, dall'altro la sua presenza costante nel campo visivo e la coerenza con lo schema funzionale chirurgico suggeriscono una funzione di supporto. Questo evidenzia come i parametri cinematici, pur fornendo informazioni fondamentali sulla dinamica del movimento, non siano da soli sufficienti a determinare il ruolo funzionale dello strumento, ma debbano essere integrati con l'analisi qualitativa del contesto operativo.

Nel complesso, i risultati indicano che la fase di esposizione della colecisti è caratterizzata da un equilibrio tra stabilità e dinamicità del gesto, in cui una pinza garantisce la

continuità della trazione e l'esposizione del campo, mentre l'altra rimane pronta a svolgere un ruolo più attivo nelle fasi successive della procedura.



	caso 1	caso 4
TotalPath (px)	5537,96	5518,61
Mean Accel (px/s^2)	1139,97	1180,41
Mean Jerk (px/s^3)	34151,28	38216,83
TotalTime (s)	29,7	29,9
Mean Speed (px/s)	184,89	182,37



	caso 0	caso 1	caso 2	caso 4
TotalPath (px)	8838,49	8704,91	6785,72	9165,51
Mean Accel (px/s^2)	980,36	1071,61	1310,53	1030,41
Mean Jerk (px/s^3)	19471,55	35432,83	36749,72	34150,06
TotalTime (s)	41,48	51,13333333	27,9	53,86666667
Mean Speed (px/s)	209,02	168,78	236,03	166,74

Figura 7.1 Grafici rappresentanti le metriche cinematiche durante la fase di esposizione della colecisti: pinza destra (a) e pinza sinistra (b)

7.2. Fase di isolamento

Nella fase di isolamento della colecisti emerge una chiara distinzione funzionale tra gli strumenti impiegati, che risulta coerente sia con i parametri cinematici analizzati sia con il ruolo operativo tipicamente associato a ciascun dispositivo. In particolare, la pinza può essere interpretata come uno strumento di supporto, mentre l'uncino assume il ruolo di strumento attivo nella dissezione dei tessuti.

Dal punto di vista quantitativo, la pinza presenta valori di percorso totale (TotalPath) significativamente inferiori rispetto all'uncino, con un range compreso tra circa 6.900 px e 45.300 px, a fronte di valori dell'uncino che raggiungono e superano i 150.000 px. Analogamente, il tempo totale di esecuzione risulta più contenuto per la pinza (circa 80–300 s) rispetto all'uncino (oltre 400–530 s), evidenziando come quest'ultimo sia impegnato per una porzione temporale più estesa della fase operativa. Anche i parametri dinamici, quali velocità media, accelerazione e jerk, risultano sistematicamente più elevati per l'uncino: ad esempio, la velocità media raggiunge valori superiori a 200 px/s e l'accelerazione supera frequentemente i 1000 px/s², mentre per la pinza tali valori si mantengono generalmente più contenuti.

Queste differenze quantitative riflettono una diversa modalità d'interazione con il tessuto. La pinza, pur mostrando una certa variabilità tra i casi (in particolare nei casi 1 e 4, caratterizzati da valori più elevati di percorso e tempo), mantiene una dinamica complessivamente più limitata e localizzata, compatibile con una funzione di stabilizzazione, trazione e mantenimento dell'esposizione del campo operatorio. Al contrario, l'uncino evidenzia un'attività più intensa, continua e spazialmente estesa, coerente con il suo utilizzo per la dissezione e l'isolamento dei piani anatomici.

Queste osservazioni mettono in evidenza come i parametri cinematici non si limitino a descrivere differenze di scala tra i due strumenti, ma evidenziano una vera e propria specializzazione funzionale: la pinza opera come elemento di supporto, garantendo le condizioni necessarie per l'intervento, mentre l'uncino rappresenta lo strumento principale attraverso cui si realizza l'azione chirurgica. Tale interpretazione risulta inoltre coerente con la pratica laparoscopica, in cui la mano non dominante è generalmente deputata alla stabilizzazione del tessuto, mentre la mano dominante utilizza strumenti attivi per la dissezione.

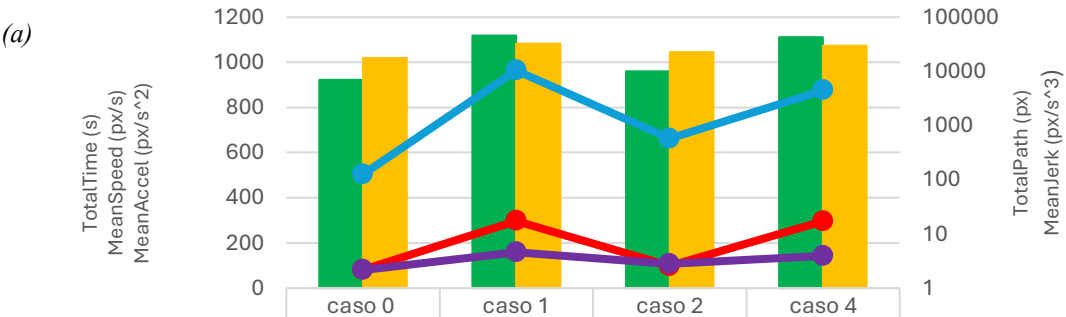
Il caso 1 e il caso 4 si distinguono per valori elevati di percorso totale e tempo di esecuzione per entrambi gli strumenti, in particolare per l'uncino, che raggiunge le percorrenze più elevate e tempi più prolungati. Questo indica una fase di isolamento più articolata e distribuita nello spazio operativo, caratterizzata da un numero maggiore di movimenti e da una dinamica complessivamente più intensa. I valori elevati di accelerazione e jerk associati all'uncino suggeriscono inoltre una dissezione più attiva e potenzialmente più complessa.

Il caso 2 presenta invece valori più contenuti di tempo e percorso, soprattutto per la pinza, e una dinamica complessiva più moderata. Questo comportamento è indicativo di una strategia più efficiente e lineare, in cui la dissezione avviene con minori aggiustamenti e una maggiore continuità del gesto. L'uncino mantiene comunque un ruolo attivo, ma con un'intensità inferiore rispetto ai casi più complessi.

Il caso 0 si configura infine come il meno impegnativo dal punto di vista cinematico, con valori ridotti di percorso, tempo e parametri dinamici per entrambi gli strumenti. Questo suggerisce una fase di isolamento più semplice o eseguita con un approccio diretto, caratterizzato da una minore estensione del movimento e da una ridotta necessità di correzioni.

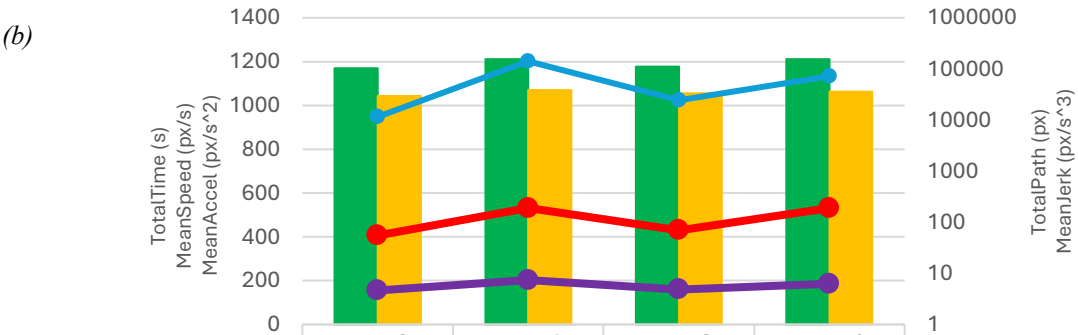
Nel complesso, anche in questa fase emerge un chiaro gradiente tra i casi: i casi 1 e 4 rappresentano configurazioni più complesse e dinamicamente intense, il caso 2 una strategia intermedia più efficiente, e il caso 0 una modalità più semplice e contenuta. Tali differenze riflettono non solo la variabilità operativa, ma anche il diverso equilibrio tra estensione del movimento, durata della procedura e intensità dinamica.

(a)



	caso 0	caso 1	caso 2	caso 4
TotalPath (px)	6907,57	45340,45	10116,25	42729,97
MeanJerk (px/s^3)	17488,1	32709,21336	22821,3	29704,64
TotalTime (s)	81,1	297,23	97,2	296,33
MeanSpeed (px/s)	79,89	158,7	106,43	141,85
MeanAccel (px/s^2)	504,02	963,87	661,73	875,2

(b)



	caso 0	caso 1	caso 2	caso 4
TotalPath (px)	104326,18	154257,9	111246,18	153964,94
MeanJerk (px/s^3)	29255,68	38498,2	33520,98	35651,69
TotalTime (s)	406,3	530,9	430,47	531,5
MeanSpeed (px/s)	155,49	202,88	160,435	185,42
MeanAccel (px/s^2)	947,95	1201,7	1023,11	1134,37

Figura 7.2 Grafici rappresentanti le metriche cinematiche della fase di isolamento: pinza (a), uncino (b)

7.3. Fase di sezione

La fase di sezione rappresenta il momento più critico e determinante dell'intera procedura chirurgica, in quanto coincide con l'esecuzione dell'atto operativo principale, ovvero la chiusura e la successiva divisione delle strutture anatomiche. In questo contesto, l'analisi combinata dei parametri cinematici e dell'osservazione qualitativa dei video consente di evidenziare con chiarezza sia la specializzazione funzionale degli strumenti sia le differenti strategie operative adottate nei diversi casi analizzati.

Dal punto di vista funzionale, emerge una netta distinzione tra il ruolo della pinza e quello degli strumenti attivi, rappresentati da clipper e forbici. La pinza si configura come uno strumento di supporto, deputato alla stabilizzazione del tessuto e al mantenimento dell'esposizione del campo operatorio. Tale funzione è confermata dai parametri cinematici, che mostrano valori relativamente contenuti di velocità media, accelerazione e jerk, associati a tempi di utilizzo più prolungati. Nei video, questo comportamento si traduce in una presenza costante e continua dello strumento, caratterizzata da movimenti lenti, controllati e poco esplorativi, finalizzati principalmente a mantenere una trazione stabile sul tessuto. La pinza, quindi, non contribuisce direttamente all'atto di sezione, ma crea le condizioni necessarie affinché gli strumenti attivi possano operare in modo efficace e sicuro.

Il clipper e le forbici, al contrario, mostrano caratteristiche cinematiche indicative di un ruolo attivo e operativamente centrale. Il clipper presenta valori intermedi di percorso totale e tempo di esecuzione, ma si distingue per valori elevati di jerk, che riflettono la presenza di movimenti rapidi e discontinui. Questo comportamento è coerente con quanto osservato nei video, dove il clipper esegue una sequenza di azioni puntuali, caratterizzate da frequenti micro-aggiustamenti necessari al corretto posizionamento delle clip. La dinamica del movimento è quindi frammentata e altamente controllata, con variazioni rapide dell'accelerazione che testimoniano la precisione richiesta in questa fase.

Le forbici, invece, mostrano una dinamica ancora più intensa, con valori elevati di accelerazione e jerk associati a tempi di esecuzione significativamente ridotti. Il percorso totale risulta più contenuto rispetto al clipper, ma concentrato in intervalli temporali brevi, indicando un'azione chirurgica rapida e localizzata. Nei video, questo si traduce in gesti decisi e mirati, in cui il taglio avviene in modo netto e puntuale, spesso preceduto da una

breve fase di posizionamento. Le forbici rappresentano quindi lo strumento principale dell'atto di sezione, caratterizzato da un'elevata intensità dinamica e da una forte concentrazione spaziale e temporale del movimento.

Un ulteriore livello di analisi è fornito dal confronto tra i diversi casi operativi (caso 0, caso 1, caso 2 e caso 4), che evidenziano una significativa variabilità nelle modalità di esecuzione della fase di sezione. Il caso 1 emerge come il più complesso, caratterizzato da valori elevati di percorso totale, tempo di esecuzione e parametri dinamici per tutti gli strumenti. Questo suggerisce una procedura più articolata, con un maggior numero di aggiustamenti e una dinamica più intensa, verosimilmente associata a una maggiore difficoltà operativa o a una strategia meno lineare. Il caso 4 presenta caratteristiche simili, ma con una distribuzione più uniforme dei parametri, indicando una gestione più controllata del gesto chirurgico, pur mantenendo un'elevata intensità.

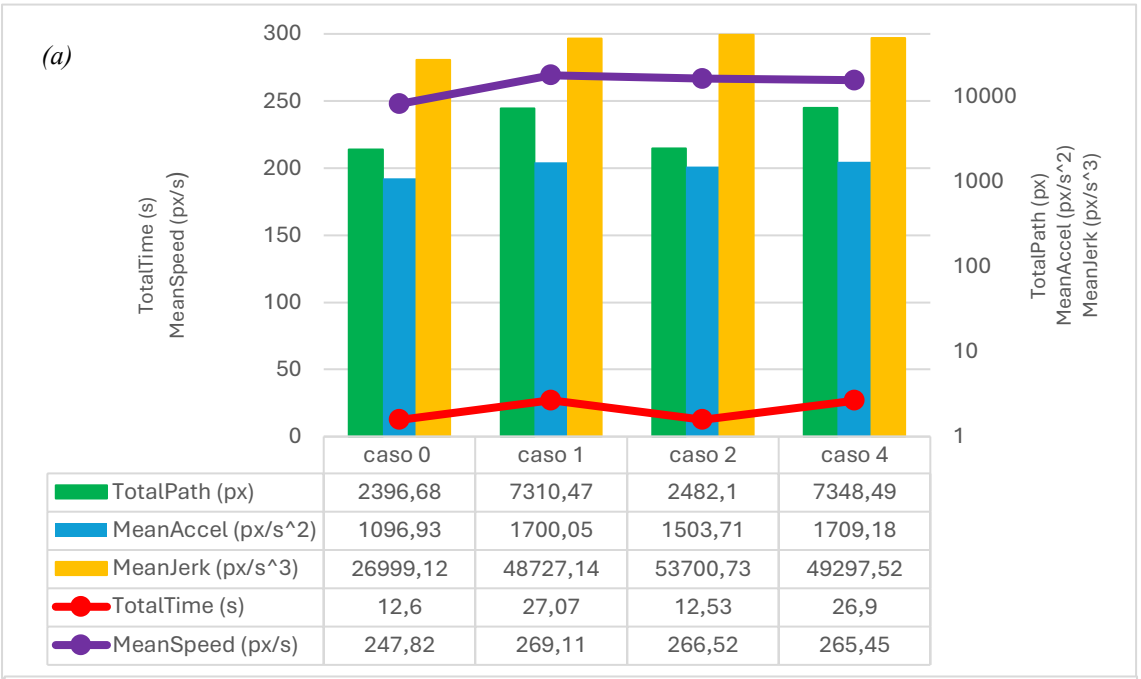
Il caso 2 si colloca in una posizione intermedia, con valori di percorso e tempo inferiori rispetto ai casi 1 e 4 e parametri dinamici più contenuti. Questo suggerisce una strategia più efficiente, caratterizzata da una maggiore linearità del gesto e da una riduzione degli aggiustamenti necessari durante l'esecuzione. Infine, il caso 0 si distingue per valori complessivamente più bassi in termini di tempo, percorso e intensità dinamica, indicando una procedura più rapida e meno complessa, o comunque eseguita con un approccio più diretto.

Il confronto tra i video conferma e rafforza queste osservazioni quantitative, evidenziando una chiara coordinazione tra gli strumenti. La pinza mantiene una trazione costante sul tessuto, stabilizzando il campo operatorio, mentre il clipper e le forbici intervengono in modo sequenziale: il primo per la chiusura delle strutture e il secondo per la loro sezione. Questa organizzazione temporale e funzionale del gesto chirurgico si riflette direttamente nei parametri cinematici, che risultano quindi non solo descrittivi della dinamica del movimento, ma anche indicativi della distribuzione dei compiti tra gli strumenti.

In conclusione, la fase di sezione appare caratterizzata da una chiara separazione tra strumento di supporto e strumenti attivi, accompagnata da una variabilità significativa tra i diversi casi in termini di complessità, durata e intensità del gesto chirurgico. L'integrazione tra analisi quantitativa e osservazione qualitativa consente quindi di delineare un quadro completo e coerente della dinamica operativa, evidenziando come la

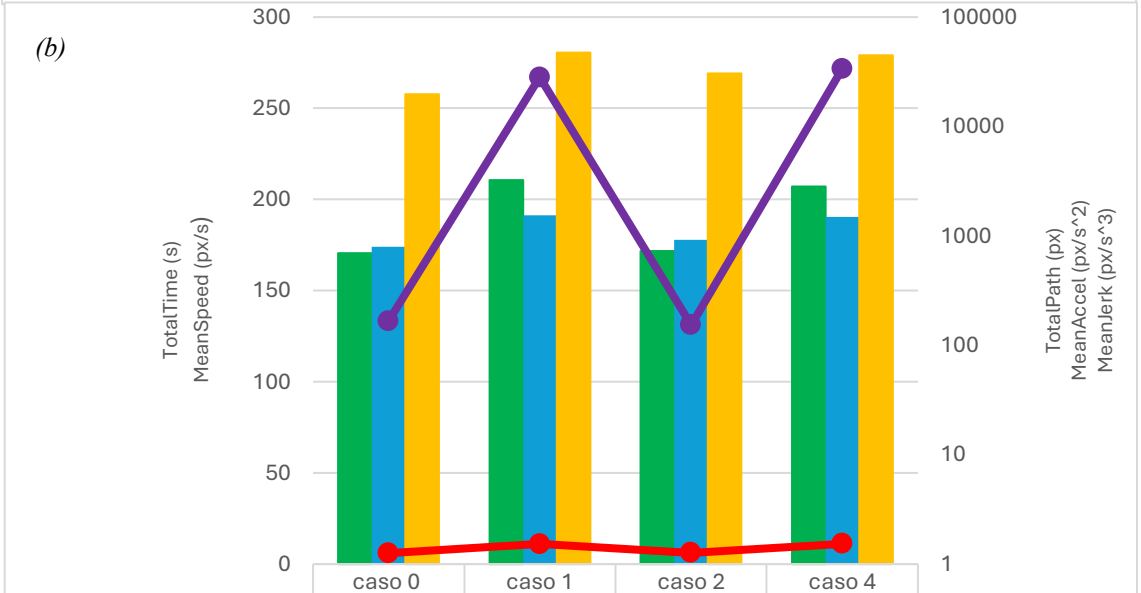
corretta coordinazione tra gli strumenti rappresenti un elemento fondamentale per l'efficacia e la sicurezza della procedura.

(a)

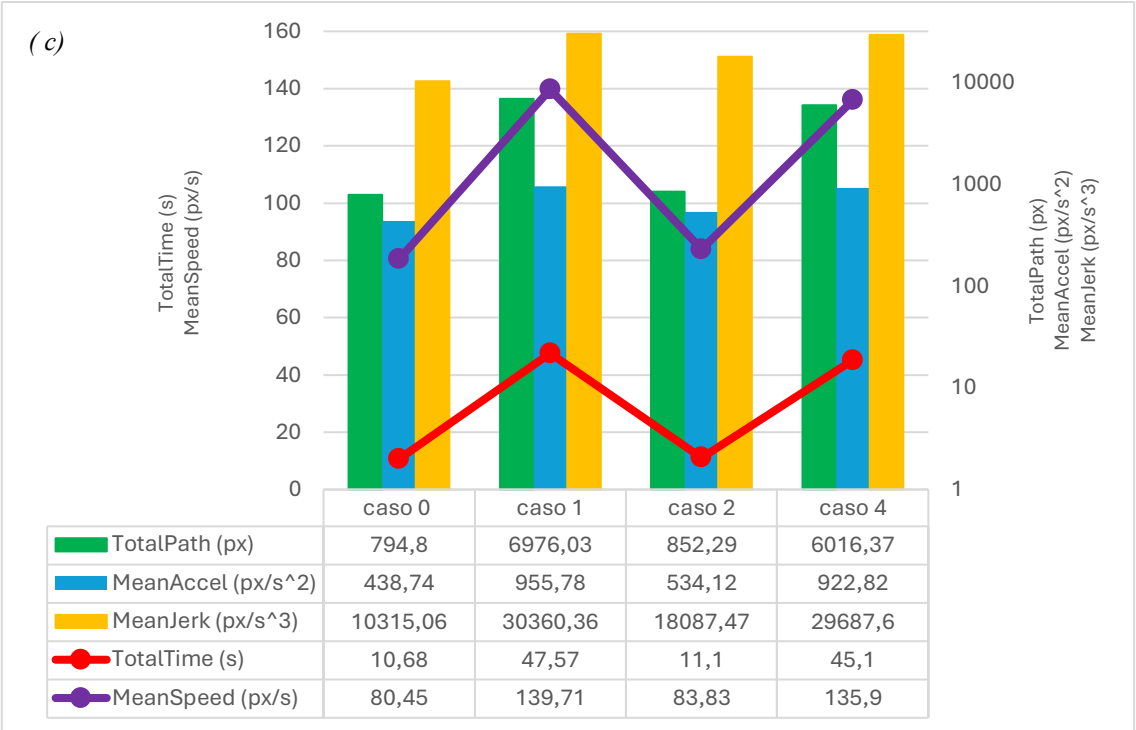


	caso 0	caso 1	caso 2	caso 4
TotalPath (px)	2396,68	7310,47	2482,1	7348,49
MeanAccel (px/s^2)	1096,93	1700,05	1503,71	1709,18
MeanJerk (px/s^3)	26999,12	48727,14	53700,73	49297,52
TotalTime (s)	12,6	27,07	12,53	26,9
MeanSpeed (px/s)	247,82	269,11	266,52	265,45

(b)



	caso 0	caso 1	caso 2	caso 4
TotalPath (px)	695,87	3220,87	730,69	2814,53
MeanAccel (px/s^2)	797,82	1545,43	916,71	1491,73
MeanJerk (px/s^3)	19831,69	47535,28	30717,91	44824,63
TotalTime (s)	6	11,07	6,2	11,1
MeanSpeed (px/s)	133,32	267,06	131,12	271,66



	caso 0	caso 1	caso 2	caso 4
TotalPath (px)	794,8	6976,03	852,29	6016,37
MeanAccel (px/s^2)	438,74	955,78	534,12	922,82
MeanJerk (px/s^3)	10315,06	30360,36	18087,47	29687,6
TotalTime (s)	10,68	47,57	11,1	45,1
MeanSpeed (px/s)	80,45	139,71	83,83	135,9

Figura 7.3 Grafici rappresentanti le metriche cinematiche fase di isolamento: clipper (a), pinza (b), forbici (c)

7.4. Fase di scollamento

La fase di scollamento è stata analizzata su un unico caso disponibile; pertanto, non è stato possibile effettuare un confronto tra diverse configurazioni operative. Tuttavia, i parametri cinematici osservati e l'analisi qualitativa del video consentono di evidenziare alcune considerazioni rilevanti in merito alla dinamica del gesto chirurgico e alla distribuzione funzionale tra gli strumenti.

In questa fase si osserva una chiara differenziazione tra i tre strumenti analizzati (grasper, hook e irrigator), che riflette il diverso ruolo operativo all'interno della procedura. In particolare, l'hook presenta valori di percorso totale (circa 163.600 px) e tempo di esecuzione (oltre 58.000 s) significativamente superiori rispetto agli altri strumenti, indicando un'attività prolungata e spazialmente estesa. Questo comportamento è coerente con il ruolo dello strumento nella dissezione dei piani anatomici durante lo scollamento,

che richiede movimenti continui e distribuiti lungo l'area di intervento. I valori elevati di accelerazione (circa 1610 px/s²) e jerk (oltre 48.000 px/s³) confermano inoltre una dinamica intensa, caratterizzata da variazioni frequenti del movimento.

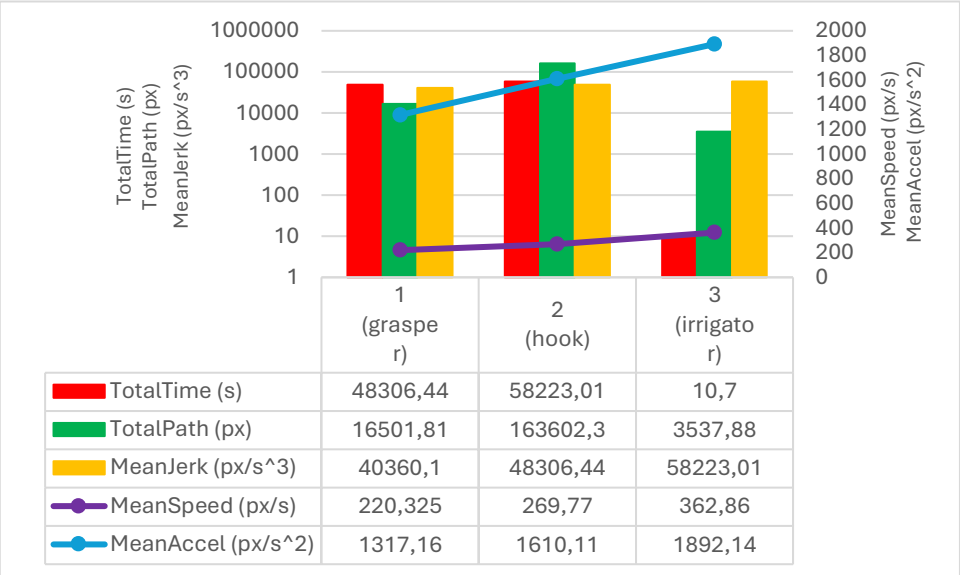
Il grasper mostra valori inferiori di percorso (circa 16.500 px) e tempo, ma mantiene comunque una presenza significativa nella fase operativa, con una velocità media e parametri dinamici moderati. Questo suggerisce un ruolo di supporto attivo, in cui lo strumento contribuisce alla trazione e alla stabilizzazione del tessuto, ma con una partecipazione più limitata rispetto all'hook in termini di estensione del movimento.

L'irrigator, infine, si distingue per un comportamento differente: pur presentando valori relativamente contenuti di percorso totale (circa 3.500 px) e un tempo di utilizzo molto ridotto (circa 10 s), mostra i valori più elevati di velocità media (oltre 360 px/s) e accelerazione (circa 1890 px/s²). Questo indica un utilizzo puntuale e rapido, caratterizzato da interventi brevi ma intensi, verosimilmente associati a operazioni di pulizia del campo o gestione dei fluidi, piuttosto che a una partecipazione continua alla dissezione.

Dal punto di vista qualitativo, il video evidenzia una dinamica del movimento che si colloca tra quella osservata nella fase di isolamento e quella della sezione: da un lato, l'azione dell'hook risulta continua e distribuita, con movimenti ripetuti lungo il piano anatomico; dall'altro, si osservano interventi più localizzati e rapidi da parte degli altri strumenti. La presenza del grasper garantisce una trazione costante, mentre l'irrigator interviene in modo episodico.

Nel complesso, anche in assenza di un confronto tra più casi, i parametri cinematici consentono di identificare chiaramente una specializzazione funzionale degli strumenti: l'hook rappresenta lo strumento principale dello scollamento, il grasper svolge una funzione di supporto attivo e l'irrigator interviene in modo accessorio e puntuale. La fase di scollamento si configura quindi come un momento intermedio tra la dissezione estesa

dell'isolamento e l'azione più localizzata della sezione, caratterizzato da un equilibrio tra continuità del movimento e controllo dinamico.



	1 (grasper)	2 (hook)	3 (irrigator)
TotalTime (s)	48306,44	58223,01	10,7
TotalPath (px)	16501,81	163602,3	3537,88
MeanJerk (px/s^3)	40360,1	48306,44	58223,01
MeanSpeed (px/s)	220,325	269,77	362,86
MeanAccel (px/s^2)	1317,16	1610,11	1892,14

Figura 7.4 Grafici rappresentanti le metriche fase di scollamento caso 4

7.5. Sviluppi futuri

L'adozione del sistema in ambito clinico richiede la risoluzione di alcune problematiche aperte di natura etica e normativa, tra cui l'approvazione da parte dei comitati etici e la gestione del consenso informato dei partecipanti. Tali aspetti sono fondamentali per garantire la validità e l'accettabilità del sistema in contesti reali, e rappresentano un passaggio imprescindibile per il trasferimento tecnologico dei risultati ottenuti.

Un naturale sviluppo del presente lavoro consiste nell'estensione del sistema di rilevamento e tracciamento automatico degli strumenti laparoscopici verso la definizione di una metrica quantitativa oggettiva della performance chirurgica che potrebbe essere basata sulla formulazione di score capace di sintetizzare in un unico indicatore diverse caratteristiche del movimento: la lunghezza del percorso fornisce una misura dell'efficienza spaziale, la velocità media riflette la rapidità di esecuzione, mentre il jerk rappresenta un indicatore della fluidità e del controllo del movimento [32].

L’integrazione di questa scala con un sistema automatico basato su modelli di *object detection e tracking*, come YOLOv26, permetterebbe di estrarre in maniera automatizzata le traiettorie degli strumenti chirurgici a partire da sequenze di video laparoscopiche. In particolare, il modello di rilevamento potrebbe essere utilizzato per identificare in tempo reale le posizioni degli strumenti, mentre un algoritmo di *tracking* multi-oggetto consentirebbe di ricostruirne le traiettorie temporali. A partire da tali traiettorie, sarebbe quindi possibile calcolare le grandezze cinematiche necessarie per la definizione della scala di performance, abilitando una valutazione oggettiva, continua e non invasiva dell’attività chirurgica [33].

Dal punto di vista metodologico, risulta fondamentale valutare la validità della scala proposta, con particolare riferimento alla *construct validity*, ovvero alla capacità della metrica di distinguere tra diversi livelli di competenza. A tal fine, è possibile progettare uno studio sperimentale in cui le metriche estratte vengano confrontate tra gruppi di chirurghi caratterizzati da differente esperienza, ad esempio chirurghi esperti e chirurghi in formazione. L’analisi statistica può essere articolata in più fasi: inizialmente, un’analisi descrittiva delle variabili cinematiche (L, A, J) e della scala composita, attraverso il calcolo di media e deviazione standard; successivamente, un’analisi inferenziale mediante t-test di Student per campioni indipendenti, al fine di verificare la presenza di differenze statisticamente significative tra i gruppi, adottando una soglia di significatività pari a $p < 0,05$. Ulteriori evidenze di validità possono essere ottenute attraverso l’analisi della correlazione tra la scala quantitativa e metriche qualitative di valutazione globale delle competenze chirurgiche, utilizzando il coefficiente di correlazione di Spearman, particolarmente adatto in presenza di dati ordinali [34].

Dal punto di vista applicativo, lo sviluppo di un sistema integrato che combini rilevamento, tracking e analisi cinematica rappresenta un passo fondamentale verso la realizzazione di strumenti di supporto alla formazione chirurgica. In tale contesto, YOLOv26 può costituire il modulo di base per il rilevamento robusto degli strumenti, mentre l’integrazione con algoritmi di tracking avanzati (ad esempio basati *su tracking-by-detection*) consentirebbe di mantenere l’identità degli strumenti lungo la sequenza temporale. Un ulteriore livello di elaborazione potrebbe includere moduli per il filtraggio del rumore, la stima delle traiettorie in coordinate reali e l’estrazione automatica delle

feature cinematiche, con l'obiettivo di costruire una pipeline completamente automatizzata [35,36].

I risultati attesi da tale approccio includono la definizione di una scala oggettiva e riproducibile della performance chirurgica, basata esclusivamente su dati quantitativi estratti automaticamente. Una tale metrica potrebbe essere utilizzata come benchmark di riferimento per la valutazione dei chirurghi in formazione, consentendo di monitorare i progressi nel tempo e di identificare specifiche aree di miglioramento. Inoltre, l'approccio potrebbe essere esteso a procedure chirurgiche più complesse, contribuendo allo sviluppo di sistemi di *computer-assisted surgery* orientati alla valutazione e al supporto decisionale [34,37].

Tuttavia, l'implementazione di tale sistema presenta alcune limitazioni. In primo luogo, la qualità delle metriche estratte dipende fortemente dall'accuratezza del rilevamento e del *tracking* degli strumenti; eventuali errori in queste fasi possono propagarsi nella stima delle traiettorie e influenzare la valutazione finale. In secondo luogo, la presenza di variabilità nella difficoltà dei casi clinici può introdurre un fattore di confondimento nella valutazione delle performance, rendendo necessario considerare strategie di normalizzazione o stratificazione dei dati. Ulteriori criticità riguardano il possibile *bias* di selezione del dataset, ad esempio nel caso di utilizzo prevalente di interventi elettivi, e la necessità di disporre di dataset sufficientemente ampi e rappresentativi.

Dal punto di vista computazionale e software, un aspetto rilevante riguarda la necessità di sviluppare soluzioni efficienti e scalabili, in grado di operare su sequenze video ad alta risoluzione. L'automatizzazione completa del processo rappresenta una sfida significativa, in quanto le soluzioni attuali spesso richiedono interventi manuali o semi-automatici e risultano time-consuming. In questo senso, l'integrazione di modelli di deep learning ottimizzati e l'utilizzo di architetture hardware dedicate possono rappresentare direzioni di sviluppo promettenti.

7.6. Conclusione

Il presente elaborato ha riguardato lo sviluppo e la messa a punto di un sistema basato su tecniche di computer vision e deep learning per la valutazione oggettiva delle competenze chirurgiche in chirurgia laparoscopica, collocandosi all'intersezione tra innovazione tecnologica e pratica clinica. Attraverso lo sviluppo di una pipeline automatizzata basata su approcci di *tracking-by-detection*, è stato possibile non solo rilevare e tracciare in modo robusto gli strumenti chirurgici, ma anche trasformare tali informazioni in un insieme strutturato di metriche cinematiche capaci di descrivere quantitativamente il gesto operatorio. L'analisi condotta ha evidenziato come tali metriche siano in grado di catturare differenze nei pattern di movimento tra operatori con diversi livelli di esperienza, confermando l'ipotesi che il gesto chirurgico possa essere modellato e valutato attraverso indicatori numerici oggettivi, riproducibili e comparabili.

Il lavoro contribuisce dunque al superamento dei limiti intrinseci delle metodologie tradizionali di valutazione, fortemente dipendenti dal giudizio soggettivo, proponendo un approccio *data-driven* che apre la strada a una standardizzazione dei criteri di *assessment* in ambito chirurgico. In questo contesto, la realizzazione di un dataset annotato ad hoc e l'adozione di architetture della famiglia YOLO si sono rivelate scelte metodologiche efficaci, garantendo un buon compromesso tra accuratezza, efficienza computazionale e adattabilità al dominio specifico, caratterizzato da elevata variabilità visiva e complessità operativa. Inoltre, il collegamento diretto tra tracking degli strumenti e analisi delle performance rappresenta un elemento di particolare rilevanza, in quanto consente di tradurre dati visivi in informazioni clinicamente interpretabili.

Nonostante i risultati incoraggianti, è opportuno sottolineare alcune limitazioni del presente studio, tra cui la dipendenza dalla qualità e dalla dimensione del dataset, la natura bidimensionale delle metriche estratte e la necessità di una validazione su coorti più ampie e diversificate. Tali aspetti rappresentano al contempo opportunità per sviluppi futuri, che potrebbero includere l'integrazione di dati tridimensionali, l'utilizzo di modelli multimodali e l'estensione dell'analisi a ulteriori tipologie di interventi chirurgici. In prospettiva, l'evoluzione di queste tecnologie potrebbe favorire l'introduzione di sistemi intelligenti di supporto alla formazione e alla pratica clinica, capaci di fornire feedback automatici e in tempo reale, contribuendo non solo al miglioramento delle competenze

individuali, ma anche all'innalzamento complessivo degli standard di sicurezza e qualità in chirurgia.

In definitiva, il presente lavoro si inserisce in un filone di ricerca in rapida espansione, dimostrando come l'analisi quantitativa del gesto chirurgico rappresenti una direzione promettente verso una chirurgia sempre più basata su evidenze oggettive, misurabili e replicabili, in cui tecnologia e competenza umana si integrano sinergicamente per ottimizzare gli esiti clinici e il percorso formativo dei futuri chirurghi.

Bibliografia

1. Luo W, Xing J, Milan A, Zhang X, Liu W, Kim TK. Multiple object tracking: A literature review. *Artif Intell*. 2021;293:103448. doi:10.1016/j.artint.2020.103448

2. Nwoye CI, Padoy N. SurgiTrack: Fine-grained multi-class multi-tool tracking in surgical videos. *Med Image Anal*. 2025;101:103438. doi:10.1016/j.media.2024.103438

3. Nwoye CI, Elgohary K, Srinivas A, Zaid F, Lavanchy JL, Padoy N. CholecTrack20: A Multi-Perspective Tracking Dataset for Surgical Tools. *arXiv*. Preprint posted online March 24, 2025:arXiv:2312.07352. doi:10.48550/arXiv.2312.07352

4. Adil Raja M, Loughran R, Mc Caffery F. A Review of Performance of Recent YOLO Models on Cholecystectomy Tool Detection. *Meas Digit*. 2025;2-3:100007. doi:10.1016/j.meadig.2025.100007

5. Moorthy K, Munz Y, Sarker SK, Darzi A. Objective assessment of technical skills in surgery.

6. Roboflow. Roboflow. https://roboflow.com/

7. Ultralytics. Ultralytics. https://platform.ultralytics.com/

8. Mascagni P, Alapatt D, Lapergola A, et al. Early-stage clinical evaluation of real-time artificial intelligence assistance for laparoscopic cholecystectomy. *Br J Surg*. 2024;111(1):znad353. doi:10.1093/bjs/znad353

9. Lin TY, Maire M, Belongie S, et al. Microsoft COCO: Common Objects in Context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, eds. *Computer Vision – ECCV 2014*. Vol 8693. Lecture Notes in Computer Science. Springer International Publishing; 2014:740-755. doi:10.1007/978-3-319-10602-1_48

10. Beck J. Quality aspects of annotated data: A research synthesis. *AStA Wirtsch-Sozialstatistisches Arch*. 2023;17(3-4):331-353. doi:10.1007/s11943-023-00332-y

11. Driscoll P. Gray's Anatomy, 39th Edition. *Emerg Med J*. 2006;23(6):492-492. doi:10.1136/emj.2005.027847

12. Hall JE, Hall ME. *Guyton and Hall Textbook of Medical Physiology*. 14th edition. Elsevier; 2021.

13. Gupta V. How to achieve the critical view of safety for safe laparoscopic cholecystectomy: Technical aspects. *Ann Hepato-Biliary-Pancreat Surg*. 2023;27(2):201-210. doi:10.14701/ahbps.22-064

14. Goodfellow I, Courville A, Bengio Y. *Deep Learning*. The MIT Press; 2016.

15. Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2016:779-788. doi:10.1109/CVPR.2016.91

16. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans Pattern Anal Mach Intell*. 2017;39(6):1137-1149. doi:10.1109/TPAMI.2016.2577031

17. Das A, Subburaj VH, Yang Y, Bednarz CW. A UAV image dataset for object detection with annotations generated using LabelImg and Roboflow. *Data Brief*. 2026;65:112483. doi:10.1016/j.dib.2026.112483

18. Szeliski R. Computer Vision: Algorithms and Applications.

19. Hussain A, Ullah K, Afaq M, Munsif M, Hussain A, Baik SW. Quality over quantity: a data-centric survey of annotation errors in object detection datasets. *Artif Intell Rev*. 2026;59(3):107. doi:10.1007/s10462-026-11502-z

20. Nassar J, Pavon-Harr V, Bosch M, McCulloh I. Assessing Data Quality of Annotations with Krippendorff Alpha For Applications in Computer Vision. *arXiv*. Preprint posted online December 20, 2019:arXiv:1912.10107. doi:10.48550/arXiv.1912.10107

21. Liao YH, Kar A, Fidler S. Towards Good Practices for Efficiently Annotating Large-Scale Image Classification Datasets. *arXiv*. Preprint posted online April 26, 2021:arXiv:2104.12690. doi:10.48550/arXiv.2104.12690

22. Redmon J, Farhadi A. YOLOv3: An Incremental Improvement.

23. Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2023:7464-7475. doi:10.1109/CVPR52729.2023.00721

24. Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2017:6517-6525. doi:10.1109/CVPR.2017.690

25. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The Pascal Visual Object Classes (VOC) Challenge. *Int J Comput Vis*. 2010;88(2):303-338. doi:10.1007/s11263-009-0275-4

26. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv*. Preprint posted online April 23, 2020:arXiv:2004.10934. doi:10.48550/arXiv.2004.10934

27. Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

28. Wang CY, Mark Liao HY, Wu YH, Chen PY, Hsieh JW, Yeh IH. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE; 2020:1571-1580. doi:10.1109/CVPRW50498.2020.00203

29. Lin TY, Dollar P, Girshick R, He K, Hariharan B, Belongie S. Feature Pyramid Networks for Object Detection. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2017:936-944. doi:10.1109/CVPR.2017.106

30. Aharon N, Orfaig R, Bobrovsky BZ. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. *arXiv*. Preprint posted online July 7, 2022:arXiv:2206.14651. doi:10.48550/arXiv.2206.14651

31. Zhang Y, Sun P, Jiang Y, et al. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *arXiv*. Preprint posted online April 7, 2022:arXiv:2110.06864. doi:10.48550/arXiv.2110.06864

32. Ganni S, Botden SMBI, Chmarra M, Li M, Goossens RHM, Jakimowicz JJ. Validation of Motion Tracking Software for Evaluation of Surgical Performance in Laparoscopic Cholecystectomy. *J Med Syst*. 2020;44(3):56. doi:10.1007/s10916-020-1525-9

33. Twinanda AP, Shehata S, Mutter D, Marescaux J, De Mathelin M, Padoy N. EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos. *IEEE Trans Med Imaging*. 2017;36(1):86-97. doi:10.1109/TMI.2016.2593957

34. Zia A, Sharma Y, Bettadapura V, et al. Automated video-based assessment of surgical skills for training and evaluation in medical schools. *Int J Comput Assist Radiol Surg*. 2016;11(9):1623-1636. doi:10.1007/s11548-016-1468-2

35. Kamtam DN, Shrager JB, Malla SD, et al. Deep learning approaches to surgical video segmentation and object detection: A scoping review. *Comput Biol Med*. 2025;194:110482. doi:10.1016/j.compbiomed.2025.110482

36. Ahmed FA, Yousef M, Ahmed MA, et al. Deep learning for surgical instrument recognition and segmentation in robotic-assisted surgeries: a systematic review. *Artif Intell Rev*. 2024;58(1):1. doi:10.1007/s10462-024-10979-w

37. Choudhry O, Ali S, Biyani CS, Jones D. Real-Time Tool Detection in Laparoscopic Datasets for Surgical Training in Low-Resource Settings. *Healthc Technol Lett*. 2025;12(1):e70045. doi:10.1049/htl2.70045