

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Biomedica

Tesi di Laurea Magistrale

***Segmentazione di immagini TC del polso e ricostruzione modello 3D:
approccio integrato con metodi tradizionali e intelligenza artificiale per
l'ottimizzazione della pianificazione preoperatoria e dell'intervento
chirurgico***



Relatori:

Prof. Filippo Molinari

Prof. Sandro Moos

Candidata

Maria Olimpia De Titta

289534

A.a. 2024/2025

1 Sommario

Abstract	4
2 Introduzione	6
3 Stato dell'arte	8
3.1 Trattamento chirurgico della frattura del polso	8
3.1.1 Struttura anatomica	8
3.1.2 Tecniche chirurgiche	12
3.2 Segmentazione di immagini mediche	14
3.2.1 Tecniche tradizionali	14
3.2.2 Tecniche basate su intelligenza artificiale	15
3.2.3 Problemi aperti	18
3.3 Applicazioni cliniche delle immagini mediche segmentate	20
4 Metodologia	21
4.1 Dataset	21
4.1.1 Informazioni generali	21
4.1.2 Distribuzione dei pazienti	23
4.1.3 Distribuzione delle fratture	24
4.2 Segmentazione manuale con algoritmo con tecniche di elaborazione di immagini	26
4.2.1 Eliminazione Gesso	27
4.2.2 Segmentazione Osso	42
4.2.3 Ricostruzione slice	49
4.3 Ricostruzione volume	51
4.4 Segmentazione con rete neurale	52
4.4.1 Training set, Validation Set, Test Set	53

4.4.2	Rete neurale	58
4.4.3	Prove effettuate	62
4.4.4	Metriche	63
4.4.5	Ricostruzione 3D	65
5	Risultati	67
6	Conclusioni e sviluppi futuri.....	84
7	Lista di figure	88
8	Lista di tabelle.....	91
9	Lista di abbreviazioni.....	92
10	Appendice.....	93
10.1	Funzioni	93
10.2	Descrizione del dataset.....	95
10.3	Codici	97
11	Bibliografia.....	100

Abstract

Attualmente, i chirurghi che operano sul polso non dispongono di una guida intraoperatoria specifica e devono usare tecnologie convenzionali, che possono rendere le procedure lunghe e invasive. Nonostante i progressi nell'imaging medico e nella navigazione chirurgica, la complessità del polso rende difficile la pianificazione e l'esecuzione precisa degli interventi.

Una segmentazione accurata del modello migliora la pianificazione preoperatoria, identificando con precisione le strutture anatomiche per una ricostruzione tridimensionale fedele dell'anatomia. Questo è essenziale per il posizionamento preciso della placca di fissaggio e delle viti, aumentando la precisione durante l'intervento chirurgico. L'adozione di modelli avanzati di segmentazione riduce anche il tempo necessario per la segmentazione.

In questo lavoro di tesi, utilizzando le immagini ottenute tramite Tomografia Computerizzata (TC), si propone l'uso di un algoritmo di segmentazione semi-automatico combinato con un modello di deep learning per ottimizzare la ricostruzione 3D della struttura anatomica del paziente, facilitando la pianificazione preoperatoria e la ricomposizione ossea durante l'intervento chirurgico. È stato sviluppato un algoritmo per la segmentazione di polso, mano, ulna e radio, contenenti anche il gesso. Si è scelto di operare sulle slice TC in vista assiale poiché offre la migliore risoluzione per garantire la qualità dell'immagine processata.

Il framework si articola in due fasi: la prima utilizza un algoritmo semi-automatico per rimuovere il gesso dalle immagini e ottenere le maschere delle segmentazioni ossee, che vengono poi importate in 3D Slicer per ricostruire il modello 3D; la seconda fase utilizza queste immagini per addestrare un modello di deep learning basato su una U-Net, una rete neurale convoluzionale, con tecniche di pre-processing, data augmentation e post-processing. I test sono stati effettuati su due set distinti di 907 immagini ciascuno, uno con immagini trattate dall'algoritmo semi-automatico e l'altro con immagini trattate da un operatore.

La qualità delle maschere ottenute è stata valutata con metriche come Dice Similarity Coefficient, Jaccard Index, sensibilità, specificità, accuratezza e Positive Predicted Value. I risultati mostrano che l'algoritmo semi-automatico rimuove correttamente il gesso, ottenendo segmentazioni comparabili ai metodi di deep learning e permettendo di ottenere modelli 3D con fratture ben riconoscibili. La combinazione dei due algoritmi semplifica il processo di segmentazione e identificazione delle fratture del polso, fornendo un supporto prezioso per migliorare la precisione e l'efficacia della guida intraoperatoria.

Un ulteriore sviluppo promettente è l'integrazione di un software di guida chirurgica basato su realtà aumentata, che può migliorare la visualizzazione in tempo reale del polso e delle aree di interesse durante l'intervento, riducendo i tempi operatori e permettendo una navigazione più intuitiva e precisa. Questo sistema può minimizzare l'invasività dell'intervento, limitando i danni ai tessuti circostanti e prevedendo meglio i risultati dell'intervento, facilitando il monitoraggio postoperatorio e riducendo il rischio operatorio.

Un ulteriore avanzamento è lo sviluppo di un sistema basato su intelligenza artificiale che determina automaticamente la configurazione ottimale dei marker per guidare il medico durante l'intervento, analizzando la frattura e suggerendo i punti su cui intervenire.

2 Introduzione

La navigazione chirurgica assistita da computer è una tecnologia di grande importanza in ortopedia, poiché migliora significativamente la precisione e la sicurezza delle procedure chirurgiche. Questa tecnologia sfrutta una combinazione di software di imaging 3D, dispositivi di localizzazione e strumenti di navigazione per assistere il chirurgo nel posizionamento preciso degli impianti ortopedici e nell'allineamento delle ossa durante le operazioni. I vantaggi della navigazione chirurgica assistita da computer includono una riduzione del rischio di errori di posizionamento degli impianti, una diminuzione della perdita di sangue, una riduzione della durata dell'intervento chirurgico e un tempo di recupero postoperatorio più breve per il paziente.

Esistono diverse tecnologie disponibili per la navigazione chirurgica assistita da computer in ortopedia, tra cui sistemi basati su immagini e sistemi basati su sensori. I sistemi basati su immagini utilizzano la tomografia computerizzata (TC) o la risonanza magnetica (RM) per creare un modello 3D dell'area operativa. Invece, i sistemi basati su sensori utilizzano sensori posizionati sui dispositivi chirurgici e sul paziente per rilevare la posizione degli strumenti chirurgici e delle ossa.

All'interno di questo contesto, la segmentazione delle immagini TC gioca un ruolo cruciale nella diagnosi e nel trattamento delle fratture ossee. Essa consente una precisa identificazione delle strutture anatomiche e delle lesioni, facilitando la pianificazione preoperatoria e migliorando gli esiti chirurgici. In particolare, nella chirurgia ortopedica, dove la complessità delle strutture ossee richiede un'elevata precisione, la segmentazione accurata è essenziale per il corretto posizionamento degli impianti e per ridurre i rischi operatori.

Tuttavia, la segmentazione delle fratture ossee rappresenta una sfida significativa a causa di vari fattori. Uno dei principali ostacoli è la presenza di gesso, che può ostruire la visione delle strutture ossee nelle immagini TC. Questa complessità rende necessario l'utilizzo di tecniche avanzate per migliorare la qualità della segmentazione e minimizzare gli errori diagnostici. Le tecniche di segmentazione manuale, sebbene accurate, sono estremamente laboriose e soggette a variabilità tra gli operatori. Al contrario, la segmentazione automatica, supportata da algoritmi di intelligenza artificiale, offre numerosi vantaggi, tra cui una maggiore efficienza, riproducibilità e capacità di gestire grandi quantità di dati con elevata precisione.

L'obiettivo principale di questo lavoro di tesi è sviluppare un sistema semi-automatico per la segmentazione delle immagini TC del polso fratturato, che includa l'automazione della rimozione del gesso e la segmentazione precisa delle ossa. Inoltre, questo sistema mira a ottenere una ricostruzione 3D del polso per facilitare la

pianificazione preoperatoria e migliorare l'efficacia delle procedure chirurgiche. La ricostruzione 3D offre infatti numerosi benefici, tra cui un significativo miglioramento nella visualizzazione dell'anatomia del paziente, la facilitazione del dialogo tra i medici e una pianificazione preoperatoria più accurata. Durante la fase intraoperatoria, la ricostruzione 3D supporta i chirurghi nel navigare con maggiore precisione, riducendo i tempi operativi e migliorando i risultati clinici.

La presente tesi di laurea magistrale si propone di analizzare e sviluppare un sistema avanzato per la segmentazione delle immagini TC e la ricostruzione 3D del polso fratturato. La realizzazione del progetto si è avvalsa di uno studio completo dello stato dell'arte, descritto nel capitolo 3, che fornisce una panoramica delle tecniche chirurgiche per il trattamento delle fratture del polso e delle metodologie di segmentazione delle immagini mediche.

Nel capitolo 4 viene presentata la metodologia adottata per sviluppare il sistema di segmentazione semi-automatico, inclusi i dettagli sul dataset utilizzato e le tecniche di pre-processing e post-processing delle immagini TC. I risultati ottenuti sono discussi nel capitolo 5, dove vengono confrontate le prestazioni delle tecniche di segmentazione manuale e automatica, valutando l'efficacia del sistema proposto tramite metriche di valutazione standard.

Infine, il capitolo 6 tratta degli sviluppi futuri di questo lavoro, esplorando le potenziali applicazioni cliniche e le direzioni di ricerca che potrebbero migliorare ulteriormente la segmentazione delle immagini TC e la pianificazione chirurgica.

3 Stato dell'arte

3.1 Trattamento chirurgico della frattura del polso

3.1.1 Struttura anatomica

Il polso è un'articolazione complessa (Figura 1) che collega le ossa dell'avambraccio a quelle della mano, consentendo movimenti come flessione ed estensione. L'uso continuativo può danneggiarlo e limitare i suoi movimenti. Specificamente, l'articolazione del polso coinvolge due ossa carpali, lo scafoide e il semilunare, che si connettono con le faccette articolari dell'estremità distale del radio.

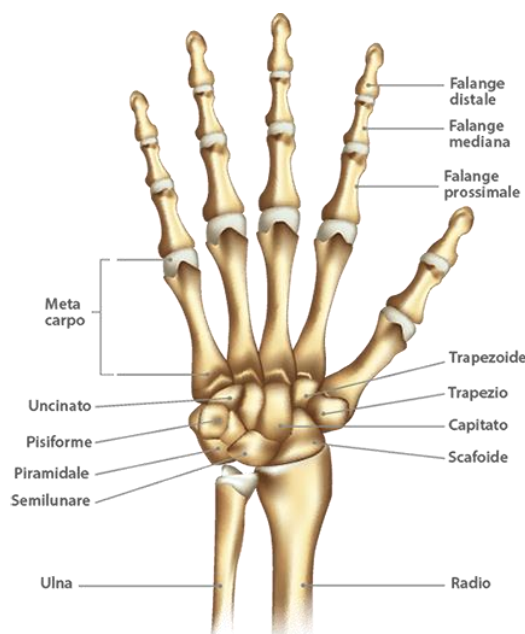


Figura 1: Anatomia articolazione del polso

Il carpo è composto da otto ossa irregolari, disposte in due file nella parte prossimale dello scheletro della mano:

- Fila prossimale: scafoide, semilunare, piramidale e pisiforme. Lo scafoide e il semilunare partecipano direttamente all'articolazione del polso.
- Fila distale: trapezio, trapezoide, capitato e uncinato.

Queste ossa carpali si articolano con i cinque metacarpi e le quattordici falangi delle dita, costituendo la parte distale dello scheletro della mano.

L'ulna e il radio sono le due ossa lunghe dell'avambraccio. Il radio, in particolare, ha tre porzioni principali:

- Estremità prossimale: si articola con l'omero per formare l'articolazione del gomito;
- Corpo: la parte centrale cilindrica che contiene il midollo osseo;
- Estremità distale: si articola con le ossa carpali scafoide e semilunare tramite due faccette articolari leggermente concave, separate da una cresta. La faccetta laterale si articola con lo scafoide, mentre quella mediale con il semilunare.

L'estremità distale del radio presenta anche due importanti elementi anatomici:

- Processo stiloideo: una proiezione ossea conica che serve come punto di inserzione per muscoli e legamenti, in particolare il muscolo brachioradiale che è uno dei principali flessori dell'avambraccio sul braccio.
- Incavo ulnare: una concavità che ospita una parte dell'ulna, contribuendo all'unione radio-ulnare.

Il polso consente diversi movimenti fondamentali per la funzionalità della mano:

- Flessione: movimento che avvicina il palmo al braccio;
- Estensione: movimento che avvicina il dorso della mano al braccio;
- Deviazione radiale: movimento che avvicina il lato del pollice al radio;
- Deviazione ulnare: movimento che avvicina il lato del mignolo all'ulna;
- Circonduzione: movimento di rotazione della mano.

Questi movimenti sono resi possibili da un complesso sistema di muscoli, legamenti e nervi. I muscoli che controllano i movimenti del polso si trovano nell'avambraccio e includono:

- Estensori del carpo: radiale breve, radiale lungo e ulnare;
 - Estensore comune delle dita;
 - Estensore lungo del pollice;
 - Flessori del carpo: radiale e ulnare;
 - Flessore superficiale delle dita;
 - Flessore lungo del pollice.
-

Questi muscoli sono collegati alla mano tramite tendini che attraversano il polso. Inoltre, tre principali nervi attraversano il polso per innervare la mano: il nervo radiale, il nervo mediano e il nervo ulnare, garantendo sia sensibilità che funzionalità motorie.

Due ulteriori movimenti del polso sono la pronazione e la supinazione. Questi ultimi sono movimenti rotatori dell'avambraccio che permettono di orientare la mano e il palmo in diverse direzioni:

- Pronazione: è il movimento che ruota l'avambraccio in modo tale che il palmo della mano sia rivolto verso il basso o all'indietro. Durante la pronazione, il radio si incrocia sopra l'ulna.
- Supinazione: è il movimento opposto alla pronazione, che ruota l'avambraccio in modo tale che il palmo della mano sia rivolto verso l'alto o in avanti. Durante la supinazione, il radio e l'ulna sono paralleli tra loro.

Uno degli infortuni più comuni che possono interessare questa complessa struttura sono le fratture del polso, le quali sono lesioni che coinvolgono le ossa dell'avambraccio distale, in particolare il radio distale, che è l'osso più comunemente fratturato in questa regione. Possono verificarsi in varie situazioni, come cadute, incidenti automobilistici o traumi diretti durante l'attività sportiva. Esistono vari modi di classificare le fratture del polso e la seguente si basa sulle caratteristiche radiografiche della frattura (Muller et al, 1993):

1. Fratture extra-articolari (tipo A), non coinvolgono le articolazioni radio-carpica né la radio-ulnare:
 - a. A1: frattura extra-articolare dell'ulna con radio integro;
 - b. A2: frattura extra-articolare semplice del radio;
 - c. A3: frattura extra-articolare pluriframmentata del radio.
2. Fratture articolari (tipo B), interessano una parte soltanto della superficie articolare:
 - a. B1: frattura sul piano sagittale del radio (stiloide);
 - b. B2: frattura del margine dorsale del radio;
 - c. B3: frattura del margine volare del radio.
3. Fratture articolari complete (tipo C), fratture in cui la superficie articolare è interrotta e separata dalla diafisi:
 - a. C1: fratture del radio articolari semplici e metafisarie semplici;
 - b. C2: fratture del radio articolari semplici e metafisarie complesse;
 - c. C3: fratture del radio articolari pluriframmentarie.

Esiste inoltre una classificazione eponimica molto diffusa di alcune tipologie di fratture:

1. Frattura di Colles: coinvolge la metafisi del radio, mantenendo intatta la superficie articolare. Nel 50% dei casi, si accompagna a una frattura dello stiloide ulnare. È il tipo di frattura del polso più comune e si verifica tipicamente a seguito di una caduta sulla mano aperta con il polso in estensione.
2. Frattura di Smith (o Colles inversa): analogamente alla frattura di Colles, la frattura di Smith interessa la stessa zona del radio distale. Tuttavia, questa frattura si verifica a causa di una caduta all'indietro con la mano aperta e il polso in flessione, provocando uno spostamento anteriore del segmento osseo fratturato. È meno frequente rispetto alla frattura di Colles.
3. Frattura di Barton: coinvolge la parte distale del radio e si caratterizza per la dislocazione dell'articolazione radio-carpica. Solitamente, si presenta in concomitanza con una frattura dello stiloide radiale.
4. Frattura di Chauffeur (o di Hutchinson): causata da una compressione dello scafoide che determina la frattura dello stiloide radiale. Avviene spesso in seguito a una caduta con il braccio esteso e la mano in deviazione esterna.
5. Frattura “Die Punch”: interessa l'osso semilunare nella regione articolare radio-carpica. È una frattura intra-articolare che può risultare da un trauma diretto sul carpo, provocando un impatto significativo sull'osso semilunare e sul radio distale.

3.1.2 *Tecniche chirurgiche*

Attualmente, il trattamento chirurgico delle fratture intra-articolari del polso si avvale di una combinazione di approcci avanzati mirati a garantire una precisa riduzione delle fratture e una stabile riparazione ossea.

Per l'accesso anteriore al polso viene frequentemente utilizzato un approccio modificato di Henry, mentre per l'accesso dorsale si adotta un approccio attraverso il terzo compartimento estensore, con il retinacolo degli estensori come supporto.

Durante l'intervento, si esegue la neurotomia del nervo interosseo posteriore nel quarto compartimento estensore, e talvolta il tendine dell'estensore lungo del pollice viene trasposto sottocutaneamente per migliorare la stabilità. Le fratture sono stabilizzate principalmente con placche di bloccaggio in titanio sia nella parte anteriore (volare) che dorsale del polso, garantendo una solida stabilizzazione dei frammenti ossei. In alcuni casi, si ricorre a innesti ossei allogenici per affrontare difetti ossei significativi e garantire un adeguato supporto durante la guarigione. La fluoroscopia intraoperatoria è ampiamente impiegata per controllare la precisione della riduzione delle fratture e l'accuratezza della posizione delle placche e delle viti, assicurando un allineamento ottimale e riducendo il rischio di complicazioni postoperatorie.

Dopo l'intervento, i pazienti sono seguiti da terapisti della mano specializzati che supervisionano un programma di riabilitazione mirato. L'immobilizzazione postoperatoria in una stecca su misura è seguita da una fase di recupero graduale che include esercizi attivi delle dita e, successivamente, del polso per promuovere la guarigione e il ripristino completo della funzionalità (Lüdi et al., 2023).

Le complicanze postoperatorie che vengono spesso riportate includono disagio o dolore dovuto all'impianto, lesione del ramo cutaneo dorsale del nervo ulnare, sindrome complessa di dolore regionale (CRPS) e infezione. Sono stati segnalati anche casi di non unione della frattura o di ridislocazione secondaria. Viene effettuata poi una valutazione clinica alla visita di controllo finale che include la misurazione dell'ampiezza di movimento del polso, la valutazione della forza di presa e di pizzicamento e la valutazione del dolore tramite scala visiva analogica (VAS). Il risultato globale viene valutato utilizzando il punteggio Quick Disabilities of the Arm, Shoulder and Hand (Q-DASH) e il punteggio del polso di Mayo. Con la radiografia viene valutata la classificazione della frattura nel pre-operatorio e la fusione ossea postoperatoria (Gauthier et al., 2022).

Un'altra tecnica innovativa impiegata per la fissazione delle fratture del polso coinvolge l'uso di guide stampate in 3D per l'inserzione percutanea di viti nello scafoide. Le guide, basate su scansioni preoperatorie del polso,

vengono sterilizzate e utilizzate durante l'intervento per garantire un'inserzione precisa delle viti con un solo tentativo, riducendo i tempi operatori e di fluoroscopia. Dopo l'intervento, il polso viene immobilizzato per 2 settimane per consentire la guarigione. Tutti i pazienti trattati con questa tecnica ottengono in genere la guarigione delle fratture entro 12 settimane, con un miglioramento significativo della funzione del polso, della forza di presa e della gamma di movimento, senza complicazioni postoperatorie come infezioni o lesioni nervose (Rong et al., 2024).

3.2 Segmentazione di immagini mediche

La segmentazione delle immagini mediche è uno strumento fondamentale per l'implementazione di tecniche avanzate a supporto della medicina. Viene utilizzata frequentemente per assistere nella diagnosi, nella pianificazione chirurgica, nel supporto intraoperatorio e nel monitoraggio post-operatorio. Inoltre, svolge un ruolo cruciale nella ricerca scientifica e nello sviluppo di nuove tecnologie mediche.

3.2.1 Tecniche tradizionali

L'introduzione delle tecniche tradizionali di segmentazione delle immagini biomediche ha rappresentato un passo importante nella chirurgia ortopedica. Metodi come la segmentazione manuale e l'utilizzo di algoritmi classici basati su soglia e contorno hanno permesso ai chirurghi di identificare le strutture anatomiche critiche con una discreta accuratezza. Queste tecniche hanno facilitato la pianificazione chirurgica e la diagnosi, offrendo un supporto per gli interventi.

Su et al., 2018 propone un metodo per la segmentazione delle ossa carpali dalle radiografie della mano, che è un compito complesso. Le sfide includono sovra-segmentazione, sotto-segmentazione e la presenza di bordi non chiaramente definiti. Un approccio efficace integra metodi basati su regioni e su confini. In particolari vengono spesso usati:

- **Thresholding Adattativo Locale:** metodo che estrae le regioni in primo piano dalle immagini, separando le ossa carpali dallo sfondo.
- **Rilevamento dei Bordi con Metodo di Canny:** fornisce una mappa dei bordi delle immagini, aiutando a identificare i confini delle ossa.
- **Integrazione delle Segmentazioni:** l'operazione XOR combina le segmentazioni delle regioni e dei bordi. Questo passaggio corregge la sovra-segmentazione e gestisce la sotto-segmentazione.
- **Raffinamento del Confine:** ulteriori metodi, come la segmentazione a controllo del watershed o i contorni attivi, migliorano ulteriormente la segmentazione integrata.

Tuttavia, le tecniche tradizionali presentano limiti in termini di precisione e tempi di elaborazione, che hanno aperto la strada alla ricerca di soluzioni più avanzate come l'intelligenza artificiale e il deep learning.

3.2.2 *Tecniche basate su intelligenza artificiale*

L'applicazione delle tecniche di intelligenza artificiale e deep learning nella segmentazione delle immagini biomediche sta rivoluzionando la chirurgia ortopedica, soprattutto negli interventi al polso. Grazie ai recenti progressi, è possibile segmentare con precisione le immagini del polso, facilitando la pianificazione e l'esecuzione delle operazioni con maggiore accuratezza. Le reti neurali convoluzionali stanno aprendo nuove frontiere nella segmentazione delle strutture anatomiche complesse, migliorando significativamente gli esiti chirurgici. L'integrazione del deep learning nella pratica chirurgica consente ai medici di navigare con maggiore sicurezza durante le operazioni, accelerando il processo diagnostico e riducendo i margini di errore, con benefici notevoli per i pazienti.

La CSR-Net (Cross-Scale Residual Network) è una rete progettata per segmentare fratture di scafoidi e ossa della mano. La rete è caratterizzata da connessioni residuali cross-scale che collegano i vari livelli della rete, migliorando la trasmissione delle informazioni e la fusione delle caratteristiche da diverse scale. La struttura della CSR-Net comprende un percorso di codifica e uno di decodifica, ciascuno con quattro livelli, che si collegano al quinto livello. Nel percorso di codifica, i dati vengono elaborati attraverso convoluzioni 3D seguite da normalizzazioni batch 3D e attivazioni ReLU. Questo approccio consente la segmentazione simultanea delle fratture di scafoidi e delle ossa della mano, assistendo sia nella navigazione chirurgica sia nella creazione di modelli digitali accurati delle fratture di scafoidi (C. Chen et al., 2021).

L'algoritmo proposto da Lee et al. utilizza una rete di deep learning basata sull'attenzione U-Net per la segmentazione delle immagini radiografiche del polso. La rete è progettata per segmentare il radio e l'ulna in immagini AP (anteroposteriori) e laterali, e identificare il lato volare/dorsale delle immagini utilizzando un modello di rilevamento e classificazione basato su RetinaNet. La struttura della rete è divisa in due percorsi principali: il percorso di codifica e il percorso di decodifica. Il percorso di codifica comprende quattro livelli in cui i dati vengono elaborati attraverso convoluzioni 3D, normalizzazioni batch 3D e attivazioni ReLU. Il percorso di decodifica è simmetrico rispetto a quello di codifica e utilizza tecniche di up-sampling per riportare le caratteristiche estratte nella loro dimensione originale, combinando le caratteristiche di basso e alto livello attraverso connessioni skip. Per migliorare le capacità di estrazione delle caratteristiche, la rete utilizza connessioni residue tra i livelli e adotta una fusione delle caratteristiche cross-scale per affrontare la ridondanza delle informazioni causata dalle connessioni residue all'interno dei livelli (Lee et al., 2024).

Un altro approccio significativo utilizza una rete neurale convoluzionale basata su una struttura U-Net simmetrica per segmentare automaticamente i contorni delle ossa carpali in immagini MRI. La CNN comprende un percorso di contrazione e un percorso di espansione, con serie di convoluzioni 3×3 , normalizzazione batch, ReLU e max-pooling 2×2 per il percorso di contrazione, e convoluzioni trasposte per aumentare la risoluzione spaziale nel percorso di espansione. Questo approccio è stato utilizzato per valutare i movimenti del polso durante l'abduzione ulnare e radiale massima continua e alternata, mostrando un'eccellente concordanza con le segmentazioni manuali dei radiologi (Radke et al., 2021).

Infine, una CNN è stata sviluppata per elaborare immagini radiografiche e identificare le fratture dello scafoide. La rete è stata addestrata su un set di dati etichettato e validata attraverso un processo di cross-validazione a cinque pieghe per assicurare robustezza e generalizzazione. La struttura della CNN include vari strati convoluzionali e di pooling, seguiti da strati completamente connessi che producono una probabilità di frattura per ciascuna immagine. L'analisi dei casi di errore ha mostrato che la CNN e i radiologi tendono a fare errori differenti, suggerendo che un approccio combinato potrebbe migliorare ulteriormente l'accuratezza diagnostica (Hendrix et al., 2021).

Le reti neurali convoluzionali, grazie alle loro capacità di segmentazione avanzate, migliorano la precisione diagnostica e riducono i tempi di analisi. Questi progressi rappresentano un passo significativo verso l'integrazione dell'intelligenza artificiale nella pratica clinica quotidiana, con potenziali benefici considerevoli per i pazienti. Tuttavia, rimangono sfide aperte, come la necessità di aumentare la quantità di dati per l'addestramento e migliorare l'integrazione clinica di questi strumenti.

3.2.2.1 Metriche di valutazione

Le metriche di valutazione dei modelli di segmentazione delle immagini mediche includono diverse misure che permettono di valutare la precisione, l'efficacia e l'affidabilità delle segmentazioni predette. Il Dice Similarity Coefficient (DSC) e l'Intersection over Union (IoU) misurano la sovrapposizione tra la segmentazione predetta e la ground truth, fornendo una valutazione della precisione spaziale delle segmentazioni. Il DSC è sensibile alle dimensioni delle regioni sovrapposte, mentre l'IoU calcola il rapporto tra l'area di sovrapposizione e l'area totale delle due regioni, fornendo una misura dell'accuratezza dell'area segmentata (Noguchi et al., 2020).

Altre metriche come il Jaccard Index e l'F1-Score forniscono una valutazione più completa considerando sia la precisione che il richiamo delle segmentazioni. Il Jaccard Index è una misura di somiglianza tra due insiemi (C. Chen et al., 2021), mentre l'F1-Score è la media armonica tra precisione e richiamo, bilanciando l'importanza di entrambe le metriche nella valutazione complessiva della segmentazione (Noguchi et al., 2020).

Inoltre, metriche come l'Accuracy e l'Area Under the ROC Curve (AUC) sono utilizzate per valutare la capacità del modello di classificare correttamente i pixel o voxel delle immagini, e la sua capacità di distinguere tra classi positive e negative. Queste metriche sono particolarmente rilevanti per valutare le performance di modelli utilizzati per la rilevazione di fratture o la segmentazione di specifiche strutture anatomiche (Lee et al., 2024).

Il confronto con i radiologi umani è cruciale per valutare l'efficacia clinica dei modelli di segmentazione. Questo confronto avviene in termini di sensibilità, specificità e AUC, confrontando le prestazioni del modello con quelle degli esperti umani nella capacità di identificare correttamente le anomalie nelle immagini mediche (Hendrix et al., 2021).

Il Positive Predictive Value (PPV) indica la probabilità che i pixel classificati come appartenenti alla regione di interesse siano effettivamente tali, fornendo una misura della precisione delle segmentazioni rispetto alla ground truth (Dalal et al., 2024).

L'applicazione clinica delle segmentazioni include la valutazione dei cambiamenti nelle larghezze dei gap scafo-lunati durante i movimenti del polso. Questo tipo di valutazione permette di monitorare la dinamica anatomica nel tempo, fornendo informazioni cruciali per la diagnosi e la gestione delle condizioni patologiche (Radke et al., 2021).

Infine, i coefficienti di correlazione di Pearson e Intraclass (ICC) sono utilizzati per misurare il grado di correlazione tra le misurazioni automatiche e manuali dei parametri radiologici. Queste misure sono importanti per valutare l'affidabilità delle misurazioni fornite dal modello rispetto a quelle effettuate manualmente da esperti radiologi (Lee et al., 2024).

In sintesi, l'uso combinato di queste diverse metriche fornisce una valutazione completa delle performance dei modelli di segmentazione delle immagini mediche, consentendo una migliore comprensione dell'accuratezza e dell'utilità clinica di tali strumenti nell'ambito della pratica medica.

3.2.3 *Problemi aperti*

Le tecniche di segmentazione di immagini biomediche presentano vari problemi aperti che devono essere affrontati per migliorare l'accuratezza, la robustezza e l'affidabilità degli algoritmi, nonché la loro integrazione nella pratica clinica.

Le immagini con distribuzioni di scala di grigi non uniformi e con bordi poco definiti rappresentano una sfida significativa per gli algoritmi di segmentazione. Inoltre, la scelta dei parametri di soglia, come quelli per il rilevamento dei bordi di Canny e per il thresholding adattativo, è cruciale e può influire notevolmente sulla qualità della segmentazione. L'integrazione post-processo richiede una buona serie di segmentazioni iniziali e potrebbe non correggere errori iniziali significativi (Su et al., 2018).

Un altro problema importante è la quantità e la variabilità dei dati. È necessario aumentare la quantità di dati per l'addestramento degli algoritmi per migliorare le loro prestazioni. Tecniche di aumento dei dati possono essere utilizzate per migliorare la robustezza dell'algoritmo, affrontando la variabilità delle immagini biomediche. Inoltre, l'aumento della risoluzione spaziale delle immagini è cruciale per ottenere una definizione più precisa delle strutture anatomiche (C. Chen et al., 2021).

L'integrazione clinica degli algoritmi di segmentazione rappresenta un'ulteriore sfida. È essenziale trovare modi efficaci per integrare questi strumenti nella pratica clinica quotidiana. Questo include affrontare la generalizzazione del modello, che richiede ulteriori dati per migliorare la capacità dell'algoritmo di funzionare

bene su dati non visti. Inoltre, è fondamentale migliorare l'affidabilità diagnostica, affrontando i casi di errore per aumentare ulteriormente l'accuratezza del modello.

Infine, è necessario affrontare i problemi di robustezza e generalizzazione del modello. Questo include l'integrazione di metodi per migliorare la capacità dell'algoritmo di funzionare bene su dati non visti e ridurre gli errori residui. L'obiettivo finale è migliorare l'accuratezza diagnostica e ridurre il tempo di analisi delle immagini rispetto ai metodi tradizionali, garantendo che gli strumenti basati su intelligenza artificiale possano essere integrati efficacemente nella pratica clinica per migliorare i risultati per i pazienti (Hendrix et al., 2021).

Per quanto riguarda le attuali tecniche utilizzate in fase operatoria, la fluoroscopia 2D convenzionale spesso non rileva il posizionamento subottimale degli impianti (viti, placche) e le incongruenze articolari durante la chirurgia delle fratture (Carelsen et al., 2008).

3.3 Applicazioni cliniche delle immagini mediche segmentate

Le immagini mediche segmentate rivestono un ruolo cruciale in diversi ambiti della pratica medica moderna. Inizialmente, supportano la diagnosi permettendo ai medici di identificare con precisione anomalie anatomiche e patologiche. Ad esempio, nella diagnosi delle fratture dello scafoide, la segmentazione delle immagini radiografiche può rivelare fratture sottili o complesse che potrebbero non essere visibili con tecniche tradizionali, migliorando la precisione della diagnosi precoce.

Durante la pianificazione chirurgica, la segmentazione offre una visualizzazione dettagliata delle strutture anatomiche. Questo supporto visivo consente ai chirurghi di pianificare interventi più precisi, compreso il posizionamento ottimale di protesi o viti, riducendo il rischio di complicazioni intraoperatorie.

Durante le procedure chirurgiche stesse, l'utilizzo di immagini mediche segmentate permette una navigazione intraoperatoria migliorata. Questa tecnologia assiste i chirurghi nell'orientamento nello spazio tridimensionale delle strutture anatomiche, garantendo un posizionamento più preciso degli strumenti e delle protesi. Ciò è particolarmente rilevante in interventi complessi dove la precisione millimetrica è cruciale per il successo dell'intervento e per il recupero ottimale del paziente.

Infine, la segmentazione delle immagini mediche gioca un ruolo essenziale nel monitoraggio post-operatorio, fornendo una valutazione dettagliata delle modifiche strutturali nel tempo. Ciò permette ai medici di valutare l'efficacia degli interventi e di rilevare eventuali complicazioni precocemente. Ad esempio, nel caso di fratture trattate chirurgicamente, la segmentazione delle immagini consente di monitorare la guarigione delle ossa nel corso del tempo, determinando se ulteriori interventi terapeutici siano necessari.

In conclusione, l'integrazione delle immagini mediche segmentate nelle pratiche cliniche quotidiane rappresenta un avanzamento significativo che migliora la precisione diagnostica, ottimizza la pianificazione e l'esecuzione chirurgica, supporta la navigazione intraoperatoria e facilita il monitoraggio post-operatorio. Questi benefici derivano dalla capacità delle tecniche di segmentazione di fornire una rappresentazione dettagliata e accurata delle strutture anatomiche, contribuendo a migliorare gli esiti clinici e a garantire cure più sicure ed efficaci per i pazienti.

4 Metodologia

L'algoritmo implementato viene descritto nel flowchart in **Error! Reference source not found.** L'obiettivo è sviluppare un algoritmo semi-automatico per rimuovere il gesso dalle immagini TC. Inoltre, deve generare le maschere dell'osso e processare le immagini per creare il dataset che sarà utilizzato per addestrare una rete neurale capace di segmentare le immagini TC del polso, dell'ulna, del radio e della mano.

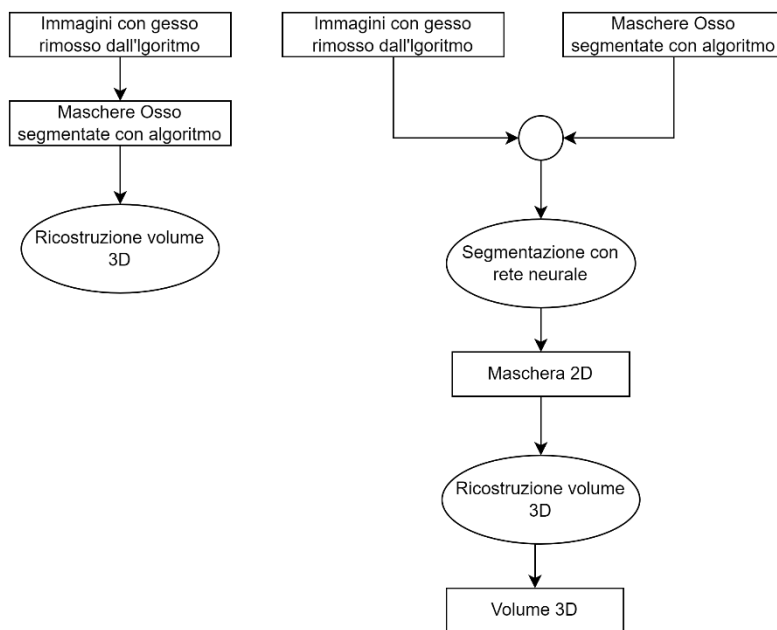


Figura 2: Flowchart generale delle metodologie

4.1 Dataset

4.1.1 Informazioni generali

Il dataset analizzato è costituito da 25 serie di immagini TC, delle quali solo due rappresentano le scansioni originali dei pazienti, mentre le restanti 23 sono viste assiali estrapolate da un software medico. Questa disparità comporta una notevole differenza nella qualità dei dati, come chiaramente visibile in Figura 3 e Figura 4.

In Figura 3 vengono confrontati i modelli 3D ottenuti dalle due diverse tipologie di serie. Il pannello a) mostra il modello 3D del polso ricostruito utilizzando esclusivamente le viste assiali estrapolate dal software medico, mentre il pannello b) presenta il modello 3D ottenuto dalla scansione originale eseguita con l'esame TC.

Nel modello a) si osserva che alcune ossa, in particolare quelle distali (più lontane dal polso), risultano tagliate o incomplete. Ad esempio, alcune falangi delle dita appaiono visibilmente più corte o addirittura mancanti rispetto al modello b). Inoltre, le articolazioni del polso non sono ben distinguibili, evidenziando una scarsa definizione strutturale. Queste problematiche sono direttamente correlate al numero di slice e allo spessore delle stesse. Nel modello a) il numero di slice è limitato a 89 con uno spessore di 2 mm ciascuna. Al contrario, nel modello b) sono presenti 542 slice con uno spessore di 0,8 mm ciascuna. La maggiore densità di slice e il minore spessore nel modello b) contribuiscono a una risoluzione più alta e a una rappresentazione più dettagliata e accurata delle strutture anatomiche.

In conclusione, la qualità del modello 3D è fortemente influenzata dal numero e dallo spessore delle slice della TC. Le scansioni originali, con un maggior numero di slice e uno spessore inferiore, forniscono una rappresentazione molto più precisa e dettagliata rispetto alle viste assiali estrapolate, che risultano in modelli parziali e meno definiti.

I software medici utilizzati per la visualizzazione delle immagini DICOM non sono progettati per mantenere le informazioni spaziali e volumetriche necessarie per una ricostruzione 3D dettagliata e precisa. Questi programmi sono utili per una visualizzazione rapida e semplice delle immagini DICOM, ma non offrono le funzionalità avanzate necessarie per l'analisi volumetrica e la ricostruzione 3D, che invece sono presenti in software come 3D Slicer. Per questo motivo, le viste assiali estrapolate dalle scansioni originali in 3D Slicer presentano un'ottima risoluzione e mantengono inalterato il numero di slice, a differenza di quelle già fornite dagli altri software.

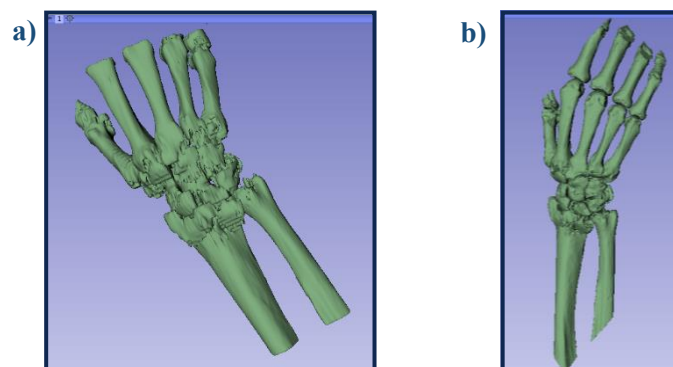


Figura 3: Modello 3D del paziente 1. a) vista assiale; b) scansione completa

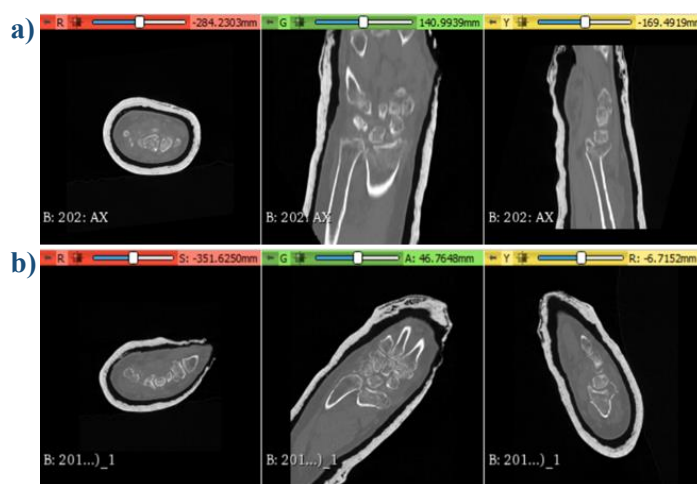


Figura 4: Viste scansione completa e assiale del soggetto 1 in 3D Slicer

4.1.2 Distribuzione dei pazienti

Il campione di soggetti analizzati costituito da 25 pazienti totali è composto dal 53% di maschi e il 47% di femmine (Figura 5).

In entrambi i casi, la fascia di età predominante è quella compresa tra 50 e 60 anni. Nello specifico, tra gli uomini vi sono principalmente soggetti di età compresa tra 20-30 anni e 40-70 anni, mentre tutte le donne, ad eccezione di una, hanno un'età superiore ai 50 anni. I dettagli completi dei dati sono riportati in Appendice: Descrizione del dataset.

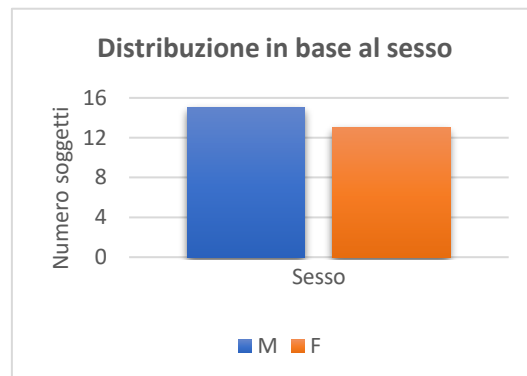


Figura 5: Distribuzione soggetti in base al sesso

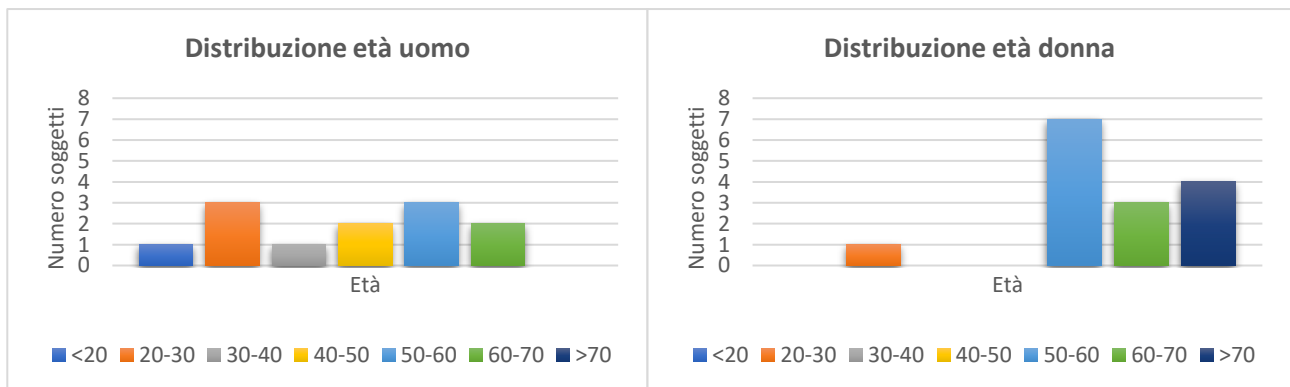


Figura 6: Distribuzione soggetti per età

4.1.3 Distribuzione delle fratture

I pazienti inclusi nel dataset presentano vari tipi di fratture (Figura 7):

- 14 con frattura del radio distale;
- 5 con frattura dell'epifisi del radio
- 2 con frattura pluriframmentaria non specificata dai clinici
- 1 con frattura articolare pluriframmentaria scomposta dell'epifisi distale del radio;
- 1 con frattura articolare pluriframmentaria scomposta articolare dell'epifisi distale del radio e frattura composta testa-ulna;
- 1 con frattura articolare pluriframmentaria scomposta dell'epifisi distale del radio e dell'ulna;

- 1 con frattura del radio distale e diastasi dello spazio scafo-lunare;
- 1 con frattura del radio distale e stotide ulnare;
- 1 con frattura scafoide, radio sinistro;
- 1 con frattura dell'epifisi distale.

I dettagli completi dei dati sono riportati nella Descrizione del dataset (vedi Appendice).

Si evince che il 50% dei soggetti presenta frattura del radio distale, il 18% frattura dell'epifisi distale e per il 7% la frattura non è stata specificata.

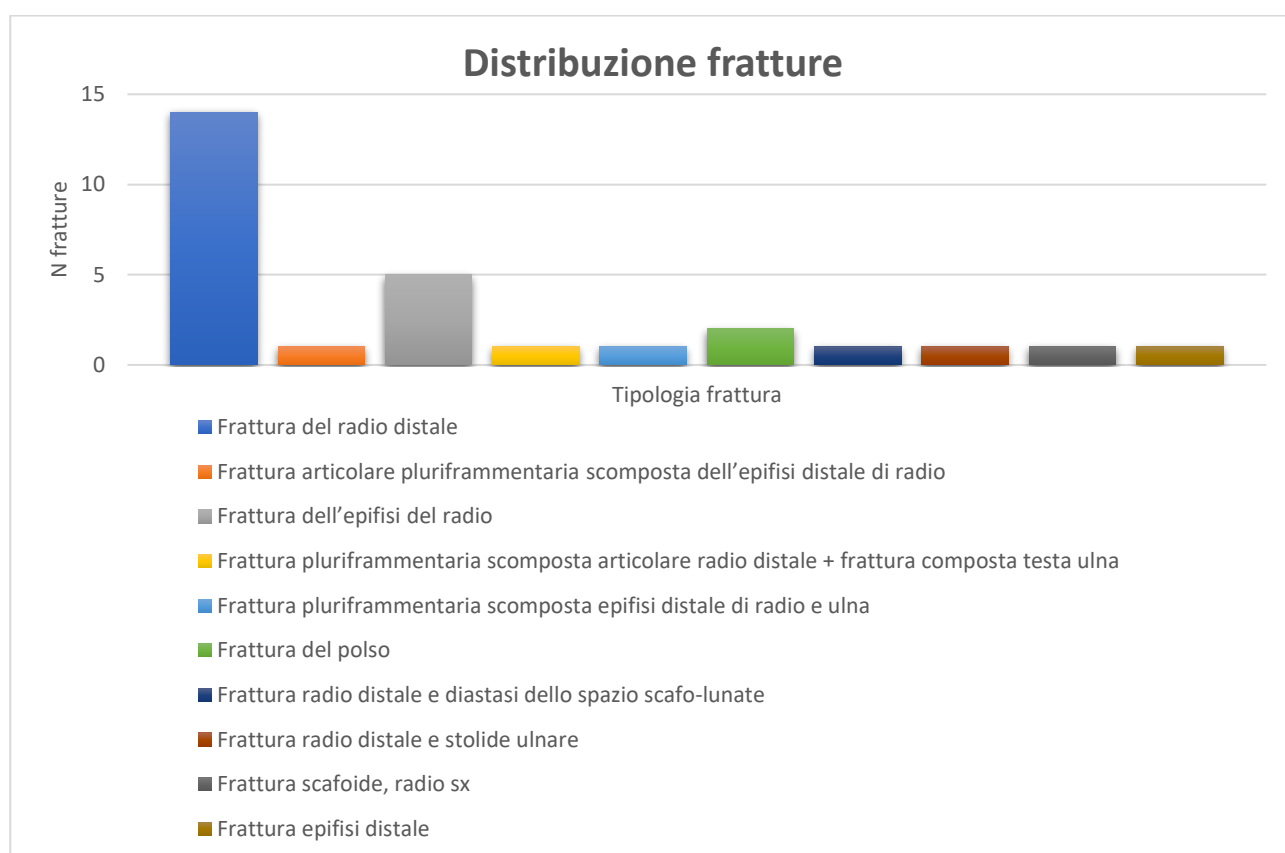


Figura 7: Distribuzione fratture all'interno del dataset

4.2 Segmentazione manuale con algoritmo con tecniche di elaborazione di immagini

L'algoritmo di segmentazione manuale è illustrato in Figura 8. Questo si articola in diverse fasi:

1. *Data preparation;*
2. *Pre-processing;*
3. Rimozione gesso;
4. Segmentazione;
5. Ricostruzione slice;
6. Ricostruzione volumi.

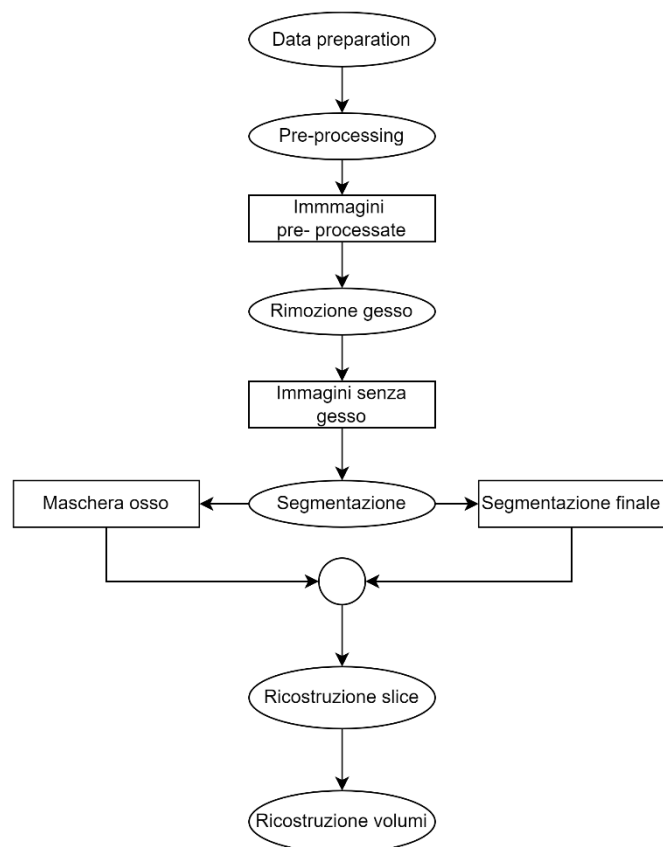


Figura 8: Flowchart segmentazione manuale con algoritmo con tecniche di elaborazione delle immagini

4.2.1 Eliminazione Gesso

La rimozione del gesso dalle immagini TC è un processo fondamentale per isolare l'osso, facilitando così la segmentazione. Questo è particolarmente importante perché l'osso e il gesso presentano intensità di pixel simili, rendendo difficile per gli algoritmi di segmentazione discriminarli accuratamente. Il processo di eliminazione del gesso illustrato in Figura 9 si articola in tre fasi principali:

1. Preparazione dei Dati (*Data Preparation*): è il primo passo fondamentale per assicurare che le immagini TC siano pronte per l'elaborazione successiva;
2. Pre-elaborazione dei Dati (*Data Pre-processing*): migliora la qualità delle immagini per l'applicazione degli algoritmi successivi;
3. Rimozione del Gesso (*Cast Removal*)



Figura 9: Workflow generale algoritmo di eliminazione gesso

La corretta eliminazione del gesso dalle immagini TC è essenziale per ottenere una segmentazione accurata dell'osso. Seguendo un processo strutturato è possibile migliorare significativamente la qualità delle immagini segmentate e la pianificazione di trattamenti.

4.2.1.1 Data Preparation

In Figura 10 è mostrato il flowchart del processo di *data preparation* delle immagini TC, articolato in quattro fasi. Questi passaggi vengono eseguiti utilizzando 3D Slicer e Python. Nella prima fase avviene il trattamento delle immagini 3D per poi passare alla gestione dei dati in 2D.

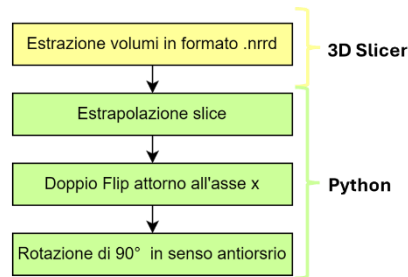


Figura 10: Flowchart Data Preparation

Slicer 3D

I file DICOM dei pazienti sono stati importati su slicer 3D (Figura 11), sono state esportate le viste in direzione assiale e salvate in formato .nrrd

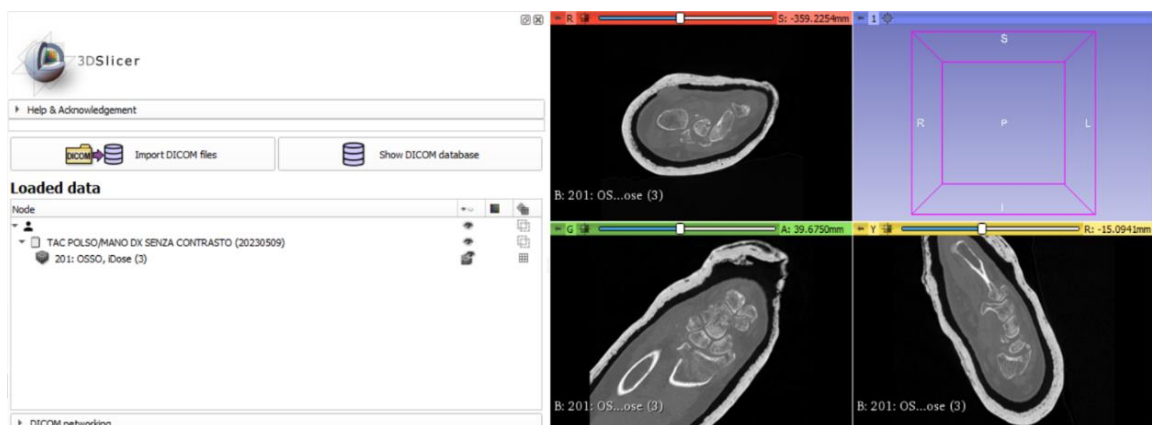


Figura 11: Import file DICOM in 3D Slicer

Python

I file in formato .nrrd sono stati successivamente importati in un ambiente IDE Python utilizzando PyCharm per l'elaborazione delle immagini. Durante il processo di importazione, è stato riscontrato un cambiamento nell'orientamento delle singole slice rispetto all'originale. Per ripristinare l'orientamento originale delle immagini, è stata applicata una doppia operazione di flip lungo l'asse x utilizzando la libreria *OpenCV* (*cv2.flip*). Successivamente, le immagini sono state ruotate di 90° in senso antiorario mediante la funzione di trasposizione di *OpenCV* (*cv2.transpose*).

Questa procedura è stata determinata confrontando l'immagine importata in Python (a sinistra in Figura 12) con la corrispondente slice originale visualizzata in 3D Slicer (a destra in Figura 12), assicurando così che l'orientamento delle immagini fosse corretto.

La scelta di utilizzare *OpenCV* è motivata dalla sua efficienza nell'elaborazione e segmentazione di immagini 2D, oltre ai tempi di elaborazione ridotti. (Kornilov & Safonov, 2018)

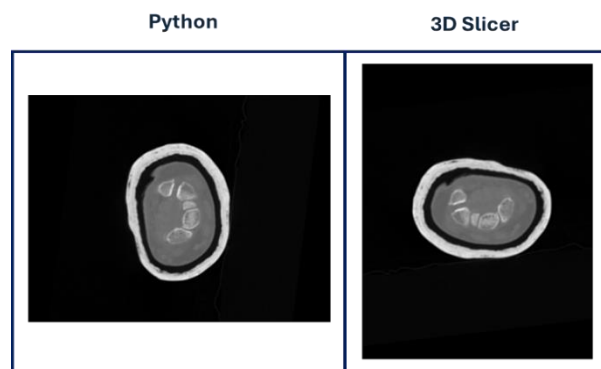


Figura 12: Confronto immagine originale - import in Python

4.2.1.2 Data Pre-Processing

Il pre-processing (Figura 13) è stato svolto interamente in Python e viene applicato sulle immagini ottenute con il *data preparation*.

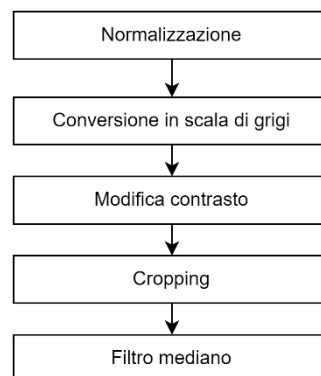


Figura 13: Flowchart Pre-Processing

1. Normalizzazione

La normalizzazione dei valori di intensità dei pixel è necessaria per rendere confrontabili le immagini TC, eliminando le variabilità dovute a fattori come il tipo di tessuto, i parametri dello strumento (energia del fascio di radiazioni, dose di esposizione) e la presenza di mezzo di contrasto.

Il metodo scelto è stato il min-max scaling, il quale scala i valori di intensità dei pixel in modo che si trovino in un intervallo comune, tipicamente tra 0 e 1. Questo è particolarmente utile per uniformare le scansioni di soggetti diversi, facilitando il confronto tra di esse.

2. Conversione in scala di grigi

In seguito alla normalizzazione, i valori dei pixel vengono moltiplicati per 255 per riportare l'immagine in scala di grigi standard, dove i valori dei pixel variano tra 0 (nero) e 255 (bianco).

3. Modifica contrasto

La conversione in uint8 ottimizza il contrasto delle immagini, garantendo una rappresentazione accurata e una gestione ottimale dei valori dei pixel durante il processo di salvataggio e visualizzazione (Lee et al., 2024).

In questo modo i valori dei pixel saranno scalati in modo appropriato per una visualizzazione accurata. Ciò assicura che l'immagine sia visualizzata correttamente su schermo e che i dettagli visivi siano mantenuti durante il salvataggio in formato PNG.

4. Cropping

Durante l'operazione di cropping delle immagini TC, viene utilizzata una funzione specifica denominata *crop_image* (vedi Appendice: Funzioni). Questa funzione, implementata manualmente in Python, consente di individuare la ROI (Region of Interest), ossia la regione dell'immagine in cui è presente la parte anatomica di interesse (Han et al., 2023). Attraverso questo processo, si ottiene una nuova immagine in cui è stata eliminata la porzione di sfondo in eccesso, contribuendo a ridurre la presenza di elementi non rilevanti. Un esempio è mostrato in Figura 14.

Tuttavia, è importante precisare che attualmente questo algoritmo non è in grado di eliminare le aree anatomiche non di interesse che presentano intensità di pixel simili a quelle dell'area di interesse. Questo potrebbe richiedere ulteriori sviluppi per migliorare la precisione del cropping.

Durante l'operazione di cropping, vengono salvate le coordinate dei vertici della ROI rispetto alle dimensioni dell'immagine originale. Questo è essenziale per ripristinare l'immagine segmentata alle dimensioni originali e per procedere con il resizing nel successivo step 5, al fine di uniformare le dimensioni di tutte le immagini.

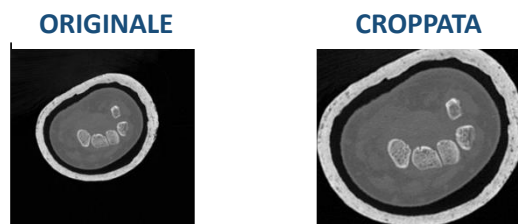


Figura 14: Confronto immagine originale e croppata

Mediante uno studio più approfondito sarebbe possibile rendere questo step completamente automatico, maggiori dettagli vengono mostrati nel capitolo Conclusioni e sviluppi futuri.

5. *Resize*

L'operazione di *resize* delle immagini *cropped* (ritagliate) è un passaggio fondamentale nel processo di ricostruzione del volume 3D, poiché assicura che ogni slice conservi le sue proporzioni originali, garantendo così l'integrità spaziale dell'arto nell'immagine finale.

Per eseguire questa operazione, si utilizza la funzione denominata *manual_resize* (vedi Appendice: Funzioni). Questa funzione è progettata in Python per adattare le dimensioni delle slice *croppate* in modo che siano tutte uniformi.

Il primo passo consiste nel confrontare le dimensioni delle singole slice *croppate* al fine di individuare le dimensioni massime sia in termini di larghezza (x) che di altezza (y). Questo confronto assicura che nessuna parte dell'immagine venga tagliata durante il *resize* e che tutte le informazioni pertinenti siano conservate.

Una volta determinate le dimensioni massime, tutte le slice *croppate* vengono ridimensionate (Figura 15) a queste dimensioni, mantenendo proporzionalità e integrità spaziale.

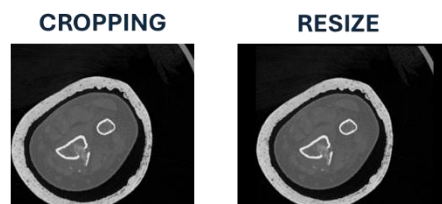


Figura 15: Confronto immagine croppata e ridimensionata

6. Eliminazione del rumore

Le immagini sono affette da rumore sale e pepe, che è un tipo di rumore che può causare il cambiamento o il ribaltamento della maggior parte dei bit di un pixel (Shengyang Luo, 2024). Questo rumore è caratterizzato dalla presenza di pixel chiari in aree scure e pixel scuri in aree chiare, creando forti differenze tra i valori di colore originali e quelli disturbati e provocando un contrasto evidente tra i pixel affetti e quelli circostanti. Il rumore sale e pepe è spesso causato da errori nella trasmissione dei dati tra il sensore della fotocamera e le unità di elaborazione delle immagini.

Per ridurre questo tipo di rumore, si utilizzano comunemente filtri mediani, che sostituiscono il valore del pixel disturbato con la mediana (o un quantile) dei valori circostanti (Staffas et al., 2024). Sebbene in letteratura il filtro mediano sia indicato come il più efficace per questo tipo di rumore, sono state effettuate anche delle prove con filtri gaussiani utilizzando diversi kernel. Tuttavia, la scelta finale è ricaduta sul filtro mediano, poiché ha dimostrato di fornire i risultati migliori.

È stato applicato un filtro mediano con kernel 9x9 (utilizzando la libreria *Pillow*, *ImageFilter.MedianFilter*) per eliminare il rumore presente sull'immagine. Questo passaggio ha migliorato la qualità complessiva dell'immagine, preservando i dettagli e i bordi principali, come si può osservare in Figura 16.

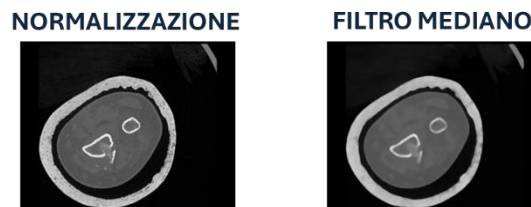


Figura 16: Confronto tra immagine normalizzata e filtrata

In Figura 17 sono mostrate delle immagini risultanti dai vari passaggi del pre-processing.

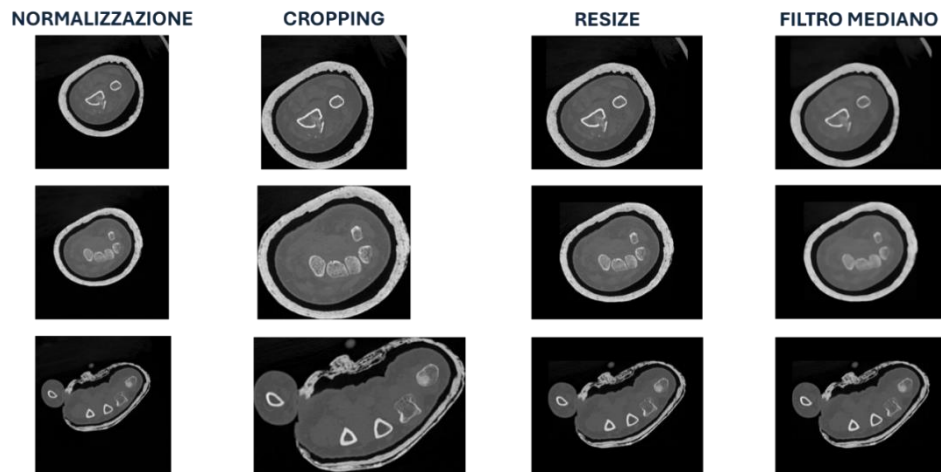


Figura 17: Immagini risultati dagli step di Pre-processing

4.2.1.3 Rimozione Gesso

Il diagramma di flusso in Figura 18 rappresenta il processo di rimozione del gesso dalle immagini TC del polso, che sono già state sottoposte a data preparation e data pre-processing.

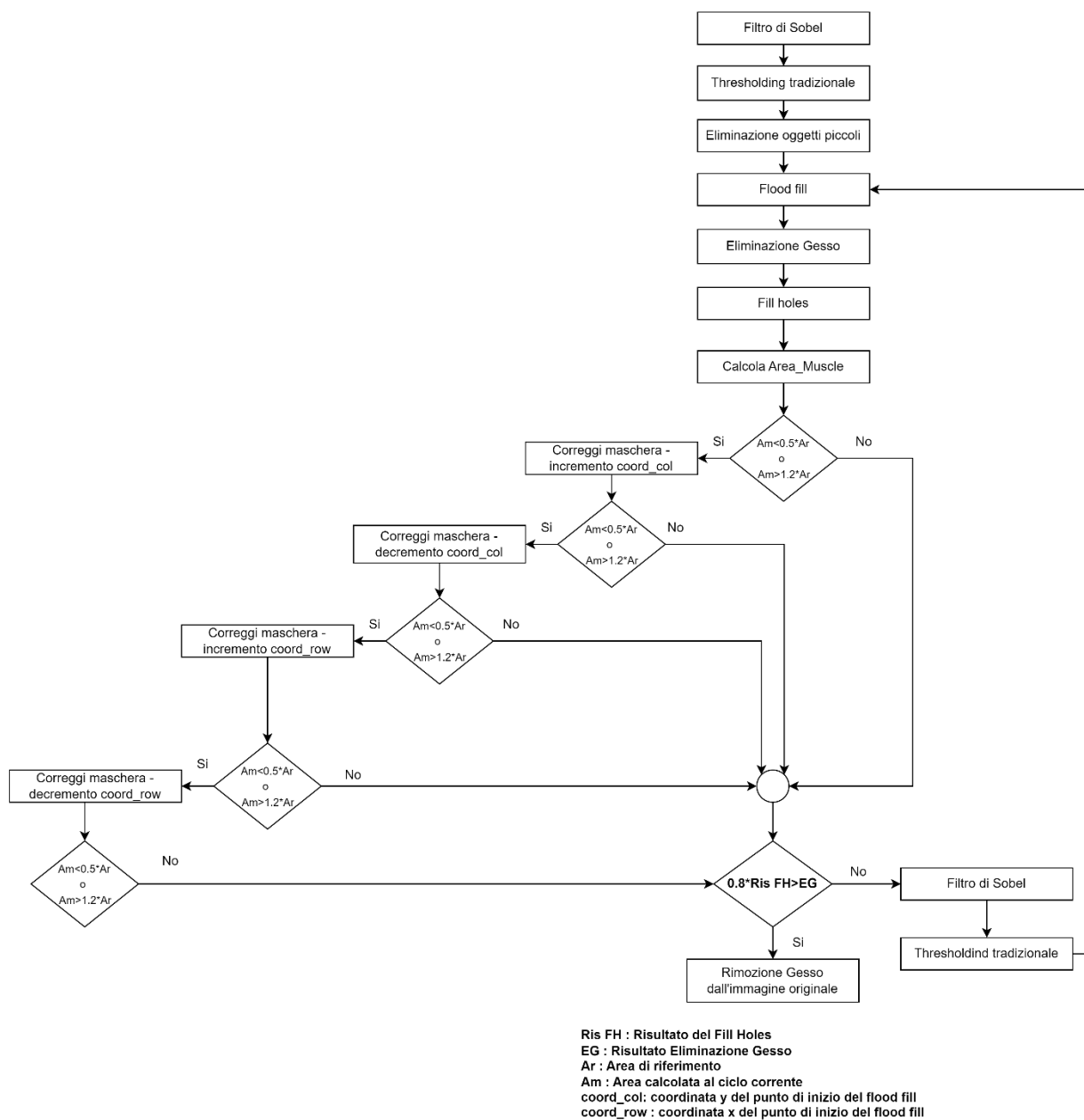


Figura 18: Flowchart processo Rimozione Gesso

Il processo segue una serie di passaggi:

1. Applicazione del Filtro di Sobel
2. Thresholding Tradizionale;
3. Eliminazione degli Oggetti Piccoli;
4. Flood Fill;
5. Eliminazione del Gesso;
6. Fill Holes;
7. Calcolo Area_Muscle
8. Controllo Area_Muscle
9. Verifica e Iterazione
10. Rimozione del Gesso dall'Immagine Originale.

In seguito, vengono dettagliati i singoli step.

1. *Filtro di Sobel*

Per individuare i bordi nell'immagine, è stato applicato il filtro di Sobel, un metodo comune per rilevare i gradienti nelle immagini, evidenziando i contorni. Questo processo coinvolge l'uso di convoluzioni con kernel predefiniti per calcolare i gradienti orizzontali e verticali, che vengono poi combinati per ottenere la magnitudine del gradiente (Vardhana et al., 2018).

Questa magnitudine evidenzia le transizioni tra diverse intensità di pixel, rivelando così i bordi nell'immagine.

I kernel di Sobel utilizzati sono i seguenti:

$$Kx = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad Ky = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

L'immagine risultante è in scala di grigi (Figura 19: Esempio applicazione filtro di Sobel. Successivamente, per ottenere la maschera binaria, è stato individuato il valore massimo di intensità dei pixel all'interno dell'immagine).

FILTRO DI SOBEL

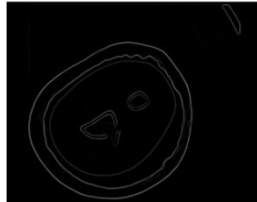


Figura 19: Esempio applicazione filtro di Sobel

2. *Thresholding*

Dopo l'applicazione del filtro di Sobel, l'immagine è stata sottoposta a una tecnica di *thresholding* tradizionale per convertirla in un'immagine binaria Figura 20: Esempio maschera ottenuta con l'applicazione del thresholding . Questo passaggio è fondamentale per separare gli oggetti di interesse dallo sfondo. Il *thresholding* è un processo di segmentazione delle immagini che comporta l'assegnazione di un valore fisso (1 per l'oggetto e 0 per lo sfondo) a ciascun pixel dell'immagine.

È stata utilizzata una soglia pari al 5% del valore massimo individuato nell'immagine. Questa soglia è stata determinata attraverso un processo di *trial and error*, in cui sono state eseguite diverse prove per selezionare quella più conservativa. Ciò è stato fatto al fine di evitare la rimozione accidentale di dettagli anatomici rilevanti.

THRESHOLDING



Figura 20: Esempio maschera ottenuta con l'applicazione del thresholding

3. *Eliminazione oggetti piccoli*

Dopo il processo di thresholding, è stato eseguito un ulteriore passaggio per rimuovere gli oggetti piccoli che non appartengono al distretto anatomico oggetto dell'analisi. Questa operazione è cruciale per ridurre il rumore nell'immagine e concentrarsi sulle strutture più significative. A tale scopo, è stata implementata manualmente una funzione denominata *KeepLargestComponent*, la cui descrizione dettagliata è disponibile nell'Appendice: Funzioni.

La dimensione minima degli oggetti da mantenere è stata impostata di default a 500 pixel, dopo una serie di prove effettuate con diverse soglie. Questa soglia è stata scelta perché ha prodotto il miglior risultato, consentendo di mantenere le strutture più rilevanti nella maggior parte delle immagini. Questo potrebbe richiedere ulteriori sviluppi per migliorare la precisione di questo step.

4. *Algoritmo di flood_fill*

Il passaggio successivo nel processo di analisi consiste nell'utilizzare il *flood fill*, una tecnica che riempie le aree interne identificate per assicurare che le regioni di interesse siano completamente delineate. Questo algoritmo individua un pixel che ha valore 0 all'interno dell'area di interesse e, a partire da quel punto, colora progressivamente tutto ciò che ha intorno, fermandosi solo quando incontra un pixel con valore 1.

Per garantire una segmentazione accurata, il punto di inizio del *flood fill* viene inizialmente impostato al centro dell'area di analisi. Tuttavia, poiché la forma e le dimensioni delle regioni di interesse possono variare tra le diverse immagini, il punto di inizio viene regolarmente spostato in base alle condizioni specifiche rilevate durante l'analisi dettagliata al punto 8.

Per implementare questo processo, è stata utilizzata la funzione *segmentation.flood_fill* della libreria OpenCV, che semplifica il riempimento delle aree interne nell'immagine in modo efficace ed efficiente.

5. *Eliminazione gesso dalla maschera di Sobel*

Dopo aver identificato e riempito le aree di interesse, si procede con l'eliminazione della porzione di gesso dalla maschera. Questa operazione viene eseguita mediante un'operazione di sottrazione booleana tra la maschera con il gesso, ottenuta tramite l'algoritmo di flood fill (come descritto nel punto 4), e la maschera ottenuta dall'applicazione del thresholding sull'immagine filtrata con il filtro

di Sobel (come descritto nel punto 2). La scelta di utilizzare questa sottrazione è stata basata su uno studio delle immagini ottenute dagli step precedenti. Questo passaggio è cruciale per isolare le strutture ossee senza interferenze esterne.

In Figura 21 è mostrato l'esempio di una slice in cui è stato effettuato il *flood fill* e successivamente l'eliminazione del gesso.



Figura 21: Esempio immagine ottenuta con l'applicazione del *flood fill* ed eliminazione gesso

6. Riempimento

Dopo aver rimosso il gesso dalla maschera mediante l'operazione di sottrazione, è importante garantire che le strutture ossee siano completamente ricostruite. A tal fine, viene eseguita un'operazione di riempimento dei buchi alla maschera risultante ottenuta nello step precedente applicando la funzione *fill_holes* del modulo *ndimage* appartenente alla libreria *Scipy* che fornisce funzionalità per l'elaborazione di immagini.

Lo scopo di questa operazione è ottenere una maschera completa e continua (Figura 22) che, una volta applicata all'immagine originale, permetta di isolare solo le porzioni interne al gesso, evidenziando così le strutture ossee senza inclusioni o aree vuote.

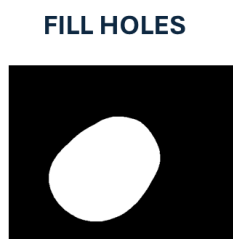


Figura 22: Esempio immagine ottenuta con applicazione di *fill holes*

7. *Calcolo Area_Muscle*

Per verificare se la segmentazione della parte interna del gesso è corretta, è necessario calcolare l'area interna risultante. Questo si fa sommando tutti i valori dei pixel della maschera segmentata. Essendo questa un'immagine binaria, in cui i pixel della regione segmentata hanno un valore 1, mentre tutti gli altri hanno un valore 0, dall'operazione di somma si ottiene il numero totale di elementi che costituiscono l'area segmentata. Questo risultato permette di verificare che l'area identificata corrisponda effettivamente alla parte interna del gesso.

8. *Controllo Area_Muscle*

Si verifica se l'area del muscolo (*Area_Muscle*) sia inferiore al 50% o superiore al 120% dell'area di riferimento (*Area_Rif*).

L'area di riferimento è l'area della maschera ottenuta dal processamento dell'immagine precedente. Nel caso della prima slice del volume, come area di riferimento viene presa quella della maschera ottenuta dal thresholding effettuato dopo il filtro di Sobel.

Questa condizione stabilisce se l'area del muscolo ottenuta tramite la segmentazione è significativamente più piccola o più grande rispetto all'area di riferimento. Una differenza troppo ampia potrebbe indicare un errore nella segmentazione o un cambiamento significativo nella forma o nella dimensione del muscolo nell'immagine.

Se questa condizione è verificata, il codice regola la posizione del punto di inizio del flood fill e lo riapplica per ottenere una segmentazione più accurata.

I pixel vicini al bordo vengono gestiti effettuando un ulteriore controllo prima sulle righe e poi sulle colonne per evitare che l'algoritmo di ricerca delle coordinate possa andare in errore.

Questo processo aiuta a garantire che il muscolo venga delineato correttamente nell'immagine, migliorando così la qualità complessiva della segmentazione e dell'analisi dell'immagine.

9. *Verifica e iterazione*

Dopo aver completato tutti i controlli sulla segmentazione, viene eseguita un'ulteriore valutazione dell'area della maschera ottenuta allo step 5. Se quest'ultima è maggiore dell'80% dell'area della maschera del muscolo (*Area_Muscle*), viene riapplicato l'algoritmo di segmentazione, ma questa volta senza eseguire la rimozione degli oggetti piccoli (step 3). Questo approccio è adottato poiché in alcune

immagini possono essere presenti dettagli disconnessi dal resto della maschera, come nel caso dei mignoli in alcune scansioni. Tali dettagli, se presentano bordi discontinui, potrebbero non essere riconosciuti correttamente dall'algoritmo di riempimento e quindi rimanere piccoli ed essere eliminati.

Questo metodo non garantisce una segmentazione ottimale di tutti i soggetti, ma è stato verificato che nei casi in cui sono soddisfatte queste condizioni specifiche, la segmentazione viene gestita in modo più accurato.

Nella Figura 23 sono mostrati due esempi di slice, evidenziando la differenza tra la segmentazione ottenuta nello step 1 e quella ottenuta nello step 2. Nel primo caso, la segmentazione risulta incompleta, mentre nel secondo caso si ottiene una segmentazione molto più vicina alla morfologia originale del soggetto. In particolare, la maschera "MaskMuscle" rappresenta l'output ottenuto dopo lo step di eliminazione del gesso, mentre la maschera "Muscolo" è quella ottenuta dopo l'operazione di riempimento dei buchi (*fillHoles*).

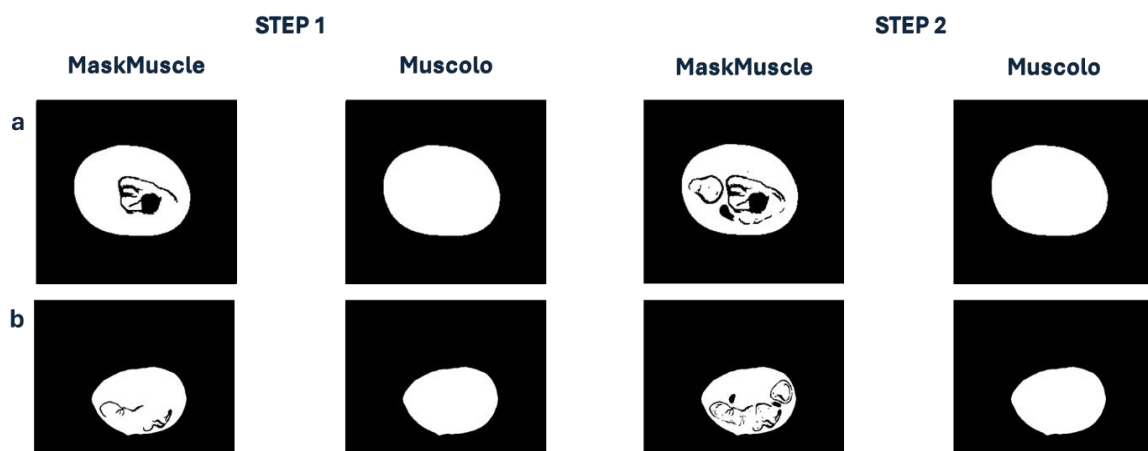


Figura 23: Confronto tra le maschere ottenute dallo step 1 e lo step 2 del processo di rimozione gesso

10. Rimozione Gesso dall'immagine originale

Infine, l'informazione ottenuta dai passaggi precedenti viene utilizzata per rimuovere il gesso dall'immagine originale. Per fare ciò, si applica una moltiplicazione tra l'immagine originale, convertita nel formato uint8, e la maschera binaria ottenuta precedentemente. La conversione al formato uint8 è necessaria affinché entrambe le immagini siano nello stesso formato, permettendo una moltiplicazione che produca il risultato corretto. È mostrato un esempio in Figura 24. Questo procedimento consente di ottenere un'immagine pulita, pronta per l'analisi, con i dettagli e i bordi principali delle strutture ossee ben preservati.

IMMAGINE FINALE



Figura 24: Esempio immagine finale dopo rimozione gesso

In Figura 25 è mostrato l'esempio di risultati ottenuti dall'applicazione dell'algoritmo di rimozione del gesso su tre slice del soggetto 1. In particolare, si evidenzia nel dettaglio i risultati intermedi dei vari step.

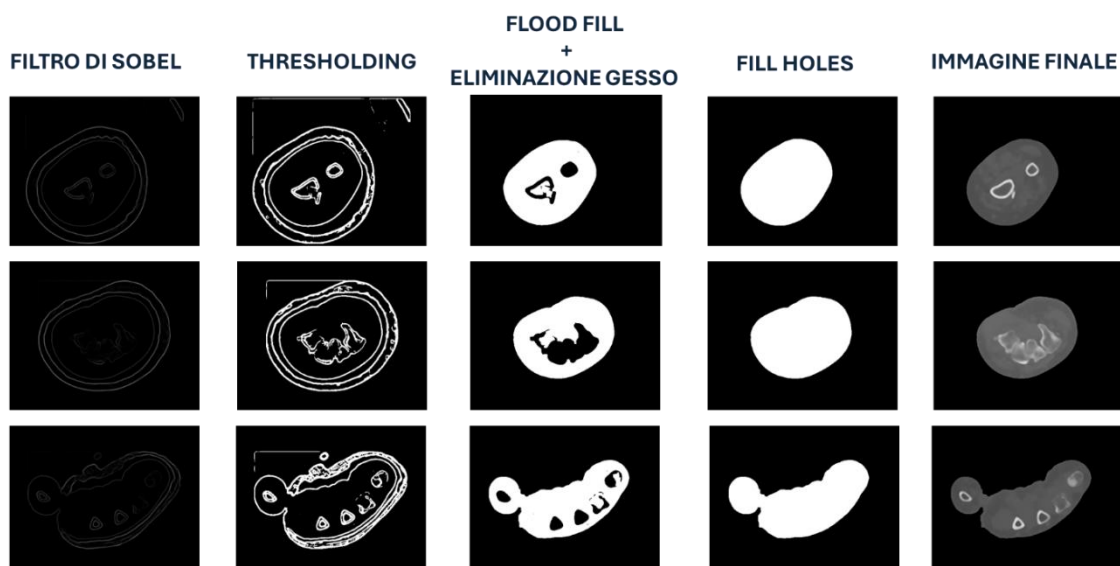


Figura 25: Rimozione gesso - Algoritmo completo

4.2.2 Segmentazione Osso

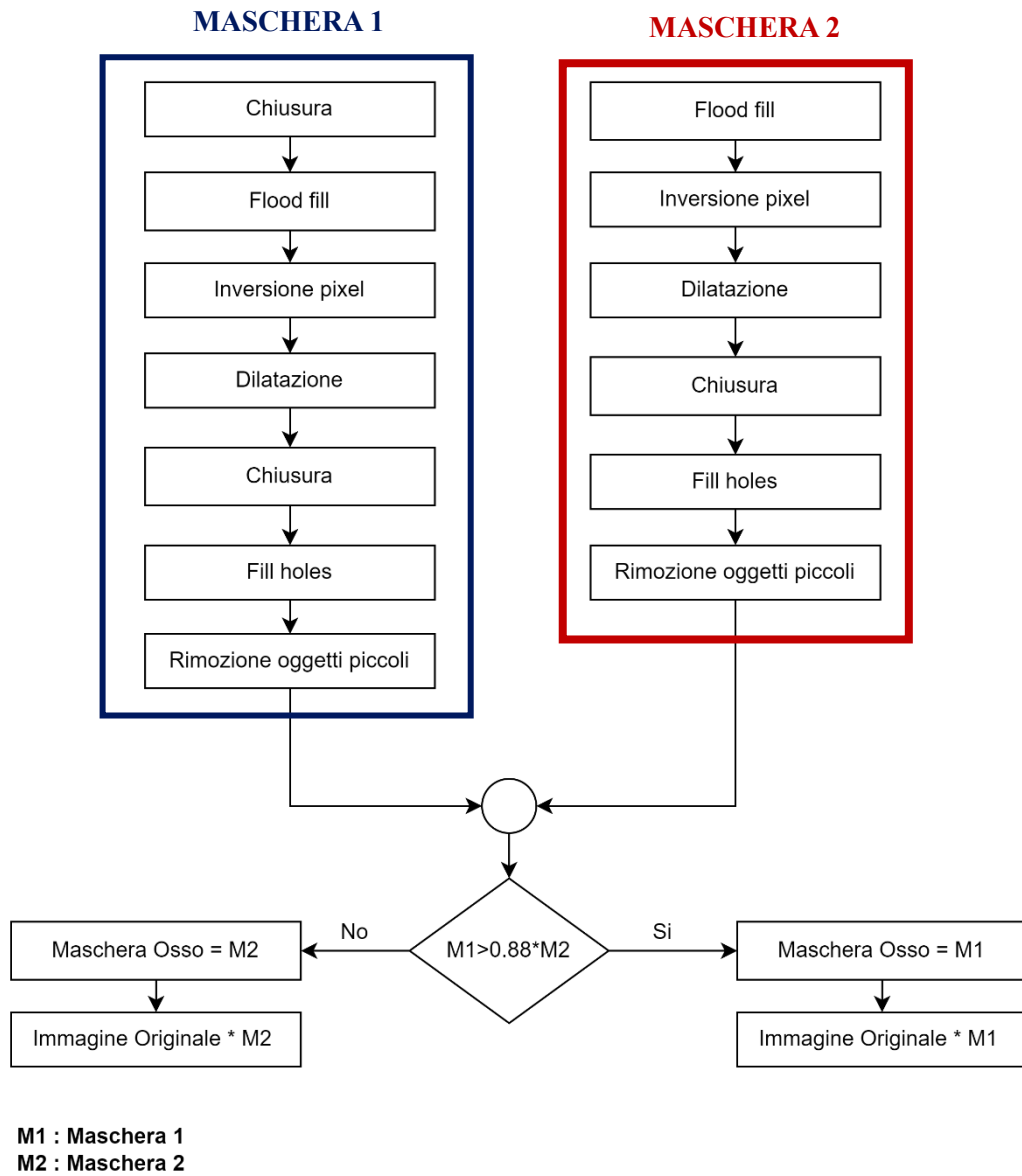


Figura 26: Flowchart processo di segmentazione dell'osso

La segmentazione dell'osso viene effettuata utilizzando le maschere ottenute nel quinto step del processo di Rimozione Gesso.

Con l'algoritmo di segmentazione osso, vengono generate due maschere differenti, denominate Maschera 1 e Maschera 2, ottenute mediante due processi che si distinguono per l'operazione di chiusura iniziale. Successivamente, viene eseguito un confronto tra le due maschere per determinare quale sia la più accurata. Una volta selezionata la maschera corretta, si procede con la segmentazione dell'immagine. In Figura 26 viene illustrato l'algoritmo completo.

Maschera 1

1. Chiusura

In preparazione al riempimento dello sfondo, che viene applicato nello step 2, si esegue un'operazione di chiusura per eliminare eventuali discontinuità nel caso in cui l'osso sia direttamente collegato allo sfondo a causa di problematiche legate allo strumento utilizzato per la scansione (Radke et al., 2021). In Figura 27 sono illustrate delle maschere ottenute da soggetti diversi in cui si verifica questa casistica. Tale operazione è necessaria per evitare che anche le parti di osso vengano colorate di bianco.

Si utilizza il comando *morphology* della libreria *OpenCV*, sfruttando un kernel 5x5.

L'operatore di chiusura è dato dalla combinazione tra dilatazione ed erosione, nello specifico erode un'immagine dilatata al fine di recuperare la forma iniziale delle strutture dell'immagine che sono state dilatate (Benediktsson et al., 2005).



Figura 27: Immagini maschere pre-chiusura

2. *Algoritmo di flood_fill*

L'algoritmo di flood fill sostituisce i pixel dello sfondo, che hanno valore 0, con pixel di valore 1 a partire da un vertice dell'immagine. Dall'analisi di tutte le maschere di partenza dell'algoritmo, è emerso che i vertici sono sempre parte dello sfondo poiché l'immagine è stata ritagliata in modo che l'oggetto fosse sempre al centro. In questo modo, si ottiene un'immagine in cui i pixel dello sfondo hanno valore 1 e quelli con valore 0 rappresentano l'oggetto. Un esempio è mostrato in Figura 28.

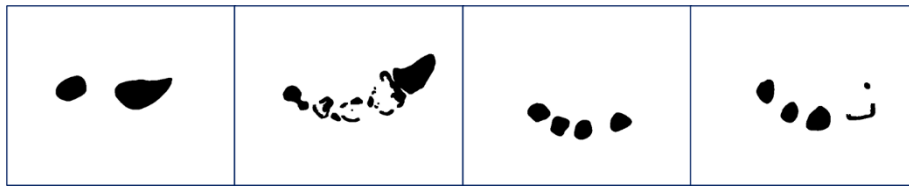


Figura 28: Esempio maschere ottenute con applicazione flood fill

3. *Inversione pixel*

Per ottenere una maschera che possa essere applicata sull'immagine originale, è necessario invertire i valori dei pixel, ottenendo come risultato finale la maschera effettiva dell'osso, in cui i pixel dello sfondo vengono impostati a 0 e quelli dell'oggetto a 1 (Figura 29).

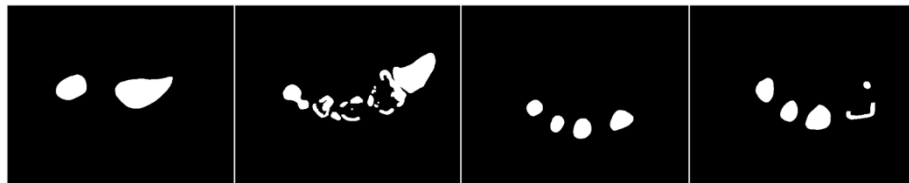


Figura 29: Esempio maschera finale

4. *Dilatazione*

Per ridurre al minimo le dimensioni dei fori ora presenti, viene applicata un'operazione di dilatazione, con un kernel quadrato 5x5.

L'operatore morfologico di dilatazione è un'operazione utilizzata nell'elaborazione delle immagini per espandere o "dilatare" le regioni degli oggetti nell'immagine (Sigmund, 2009). Questa operazione funziona confrontando ogni pixel dell'immagine con un elemento strutturale, che è un piccolo modello o pattern definito dall'utente.

Si considera il massimo valore tra i pixel nell'area definita dall'elemento strutturale. Se anche solo uno dei pixel nell'area ha valore 1 (cioè, è attivo), allora il pixel centrale dell'elemento strutturale viene impostato a 1 nell'immagine risultante. Questo significa che l'operazione di dilatazione amplifica le regioni dell'oggetto; quindi, questo processo espande o allarga le regioni dell'oggetto nell'immagine, poiché anche i pixel adiacenti all'oggetto diventano parte dell'oggetto stesso.

5. *Chiusura*

Viene eseguita nuovamente una chiusura con un kernel 7x7 per chiudere fori residui. In questo caso sono state aumentate le dimensioni del kernel per assicurarsi una maggiore efficienza in quanto con kernel più piccoli non venivano apportate modifiche (Radke et al., 2021).

6. *Fill holes*

Alla maschera ottenuta nello step precedente viene applicata un'operazione di riempimento binario dei buchi (fill_holes) per colmare le aree racchiuse dai bordi dell'osso, garantendo che le cavità interne vengano riempite. L'obiettivo è ottenere una segmentazione finale che mantenga il maggior numero di dettagli anatomici, preservando con precisione la struttura complessiva dell'osso.

7. *Rimozione oggetti piccoli*

La rimozione degli oggetti di piccole dimensioni viene applicata per eliminare i pixel spuri e l'eventuale rumore residuo. Per garantire che la morfologia della segmentazione non venga alterata, questa operazione viene eseguita tramite tre funzioni implementate manualmente, denominate rispettivamente: *Count_object_areas*, *Filtered_object_areas* e *create_filtered_binary_image*. Queste funzioni sono descritte in dettaglio nell'Appendice: Funzioni.

Le tre funzioni svolgono i seguenti compiti:

- *Count_object_areas*: Identifica gli oggetti presenti in un'immagine binaria e calcola le loro aree.
- *Filtered_object_areas*: Filtra gli oggetti identificati in base alle loro dimensioni, escludendo quelli inferiori a una certa soglia prestabilita.
- *create_filtered_binary_image*: Crea una nuova immagine binaria filtrata che include solo gli oggetti che soddisfano i criteri di dimensione definiti.

Il risultato di questo processo è una maschera binaria dell'osso che conserva esclusivamente gli oggetti desiderati, garantendo così una segmentazione pulita e accurata.

Maschera 2

La seconda maschera (maschera 2) viene ottenuta seguendo gli stessi passaggi della maschera 1, ad eccezione della prima operazione di chiusura. Dai test effettuati su diversi soggetti è emerso che, in alcuni casi, il primo algoritmo rimuoveva una parte significativa della segmentazione perché la chiusura veniva eseguita in punti errati, causando l'eliminazione di molti oggetti di interesse durante la fase di rimozione.

Pertanto, è stato deciso di generare entrambe le maschere per ogni slice e successivamente confrontarle (step 8) per determinare quale fosse quella corretta per ogni slice.

8. Confronto maschere

Per determinare quale delle due maschere sia la più appropriata per ciascuna slice, viene effettuato un confronto basato sulla somma dei pixel bianchi (che rappresentano le aree segmentate dell'osso) nelle due maschere. Se l'area della maschera 1 è maggiore del 88% di quella della maschera 2, viene scelta la maschera 1. Altrimenti, si seleziona la maschera 2.

Questa soglia del 88% è stata scelta su base sperimentale. Durante la fase di test, si è osservato che era necessario utilizzare una soglia molto alta per evitare errori di selezione, poiché alcune maschere risultavano molto simili tra loro ma presentavano differenze nei dettagli. Anche se tali dettagli erano di dimensioni poco significative, erano comunque importanti per una segmentazione accurata dell'osso. La selezione della soglia è avvenuta partendo da un valore del 70% e aumentando gradualmente man mano che venivano effettuate le prove, fino a identificare la soglia più adatta.

Questo valore sperimentale assicura che la maschera 1 venga scelta solo se contiene una quantità sostanzialmente maggiore di informazioni rispetto alla maschera 2. Altrimenti, viene preferita la maschera 2 per preservare i dettagli. Questo è dovuto al fatto che la maschera 1 include l'operazione di chiusura; quindi, sarà maggiore dell'88% rispetto alla maschera 2 solo nel caso in cui l'operazione di chiusura abbia effettivamente migliorato la segmentazione.

Inoltre, questa soglia permette di gestire i casi in cui la maschera 2 è sovrasedgmentata. In tal caso, la maschera 1 viene mantenuta poiché, nonostante una segmentazione più stringente, rispetta comunque il criterio di selezione. È importante notare che, se la maschera 2 è eccessivamente sovrasedgmentata, l'algoritmo continuerà comunque a selezionarla. Questo si potrebbe migliorare con sviluppi futuri dettagliati nel capitolo “Conclusioni e sviluppi futuri”.

Pertanto, il confronto con la soglia del 88% permette di selezionare la maschera che meglio rappresenta la struttura dell'osso per quella specifica slice, preservando i dettagli essenziali.

In Figura 30 sono presentati esempi di confronto tra le due maschere, dove M1 rappresenta maschera 1 e M2 rappresenta maschera 2.

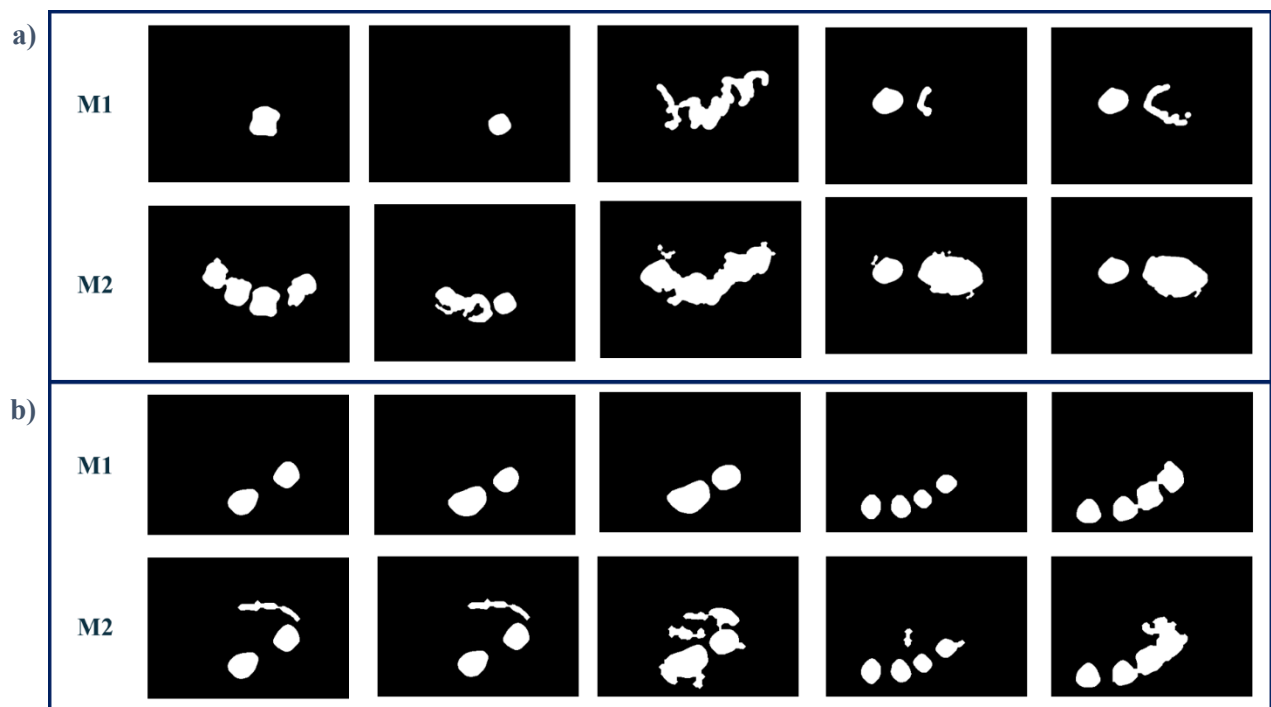


Figura 30: Confronto tra Maschera 1 e Maschera 2

Nel primo esempio (a), viene evidenziata la situazione in cui Maschera 1 < 0.88 * Maschera 2. Qui, l'operazione di chiusura ha causato la rimozione di elementi critici, compromettendo la corretta ricostruzione dell'osso e rendendo indistinguibile l'informazione relativa alla frattura.

Nel secondo esempio (b), si illustra il caso in cui Maschera 1 $>$ 0.88 * Maschera 2. Questo non è immediatamente evidente. Si ricorda che nella scelta della soglia si è tenuto conto dalla necessità di mantenere l'integrità dei dettagli nelle situazioni in cui la seconda maschera risultasse sovra-segmentata come nelle slice illustrate.

9. Segmentazione

Infine, per isolare l'osso viene eseguita un'operazione di moltiplicazione tra l'immagine originale e la maschera finale ottenuta. In questo modo si ottiene la segmentazione dell'osso che verrà utilizzata per la ricostruzione del volume e la verifica della corretta riproduzione della frattura.

Per verificare se l'algoritmo completo fosse efficace è stato osservato cosa succede se l'algoritmo si ferma allo step 3 quindi se si effettuano solo chiusura, *flood fill* e inversione dei pixel.

In Figura 31 sono presentati i risultati ottenuti dai due processi. Si osserva che l'algoritmo completo mostra un approccio più conservativo, poiché, riempiendo le aree circondate dai bordi, produce una segmentazione che permette di mantenere un livello di dettaglio più elevato. Al contrario, utilizzando l'algoritmo ridotto (step da 1 a 3), si ottengono maschere più ridotte e segmentazioni in cui mancano diverse caratteristiche.

Con l'algoritmo completo, è possibile identificare la frattura in modo più affidabile in quanto consente di verificare la sua presenza esatta nel punto evidenziato. Al contrario, con l'algoritmo ridotto, non vi è alcuna certezza poiché viene considerata solo la parte esterna dell'osso. Di conseguenza, non è possibile garantire l'accuratezza della segmentazione della frattura, poiché manca una visione completa dell'osso. Pertanto, non si può distinguere con sicurezza tra una frattura e un errore di rilevamento dell'algoritmo.

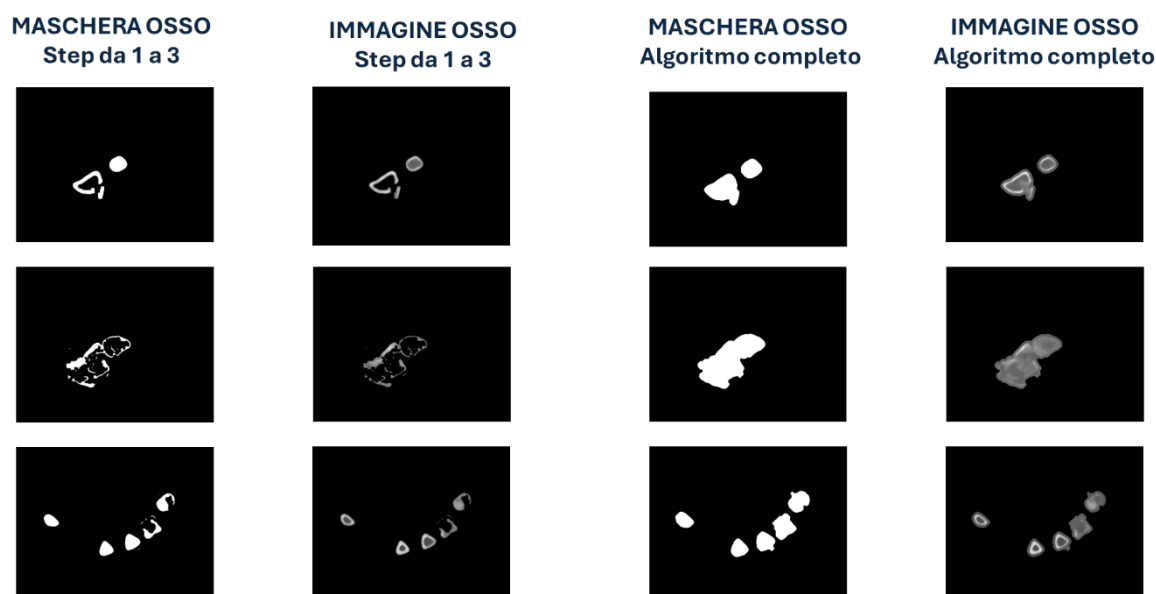


Figura 31: Confronto segmentazione ottenuta dall'algoritmo completo e ridotto

4.2.3 Ricostruzione slice

Per sviluppare un algoritmo di deep learning per la segmentazione di polso, mano, ulna e radio, le immagini pre-processate e le segmentazioni ottenute vengono riportate alle dimensioni originali, ripristinando il posizionamento dell'immagine *croppata* rispetto a quella originale. Questo passaggio è fondamentale anche per la successiva ricostruzione 3D. I casi di immagini ricostruite sono tre: slice osso, slice con gesso rimosso e slice pre-processate con gesso rimosso.

- **Ricostruzione slice osso**

Il processo prevede la ricostruzione delle slice contenenti solo la porzione di osso ottenuta dall'immagine originale. Questo avviene attraverso la creazione di una maschera dell'osso, realizzata mediante una funzione dedicata, implementata manualmente e denominata *reconstruction_osso* (Appendice: Funzioni). Tale funzione ripristina un'immagine *croppata* alle sue dimensioni originali, utilizzando le coordinate del ritaglio e i valori delle differenze tra le dimensioni originali e quelle ridimensionate.

La maschera dell'osso così ricostruita viene poi moltiplicata per le slice originali, ottenendo come risultato un'immagine che contiene solo il tessuto osseo originale. Questo procedimento viene applicato a entrambe le tipologie di immagini salvate nello step precedente.

Non vengono mostrati i risultati in quanto il contrasto originale dell'immagine non permette una visualizzazione dell'immagine chiara.

- **Ricostruzione slice con gesso rimosso**

La maschera ottenuta dopo la rimozione del gesso viene ricostruita utilizzando la funzione *reconstruction_osso*. Successivamente, questa maschera viene moltiplicata per le slice originali, ottenendo così le immagini originali senza gesso ripristinate nella loro posizione originale.

Anche in questo caso non vengono mostrati risultati per la medesima problematica indicata prima.

- **Ricostruzione slice pre-processate con gesso rimosso**

La funzione *reconstruction_osso* viene applicata alle immagini pre-processate dalle quali è stato rimosso il gesso. In Figura 32 vengono mostrati degli esempi di tale caso, prima e dopo la ricostruzione.

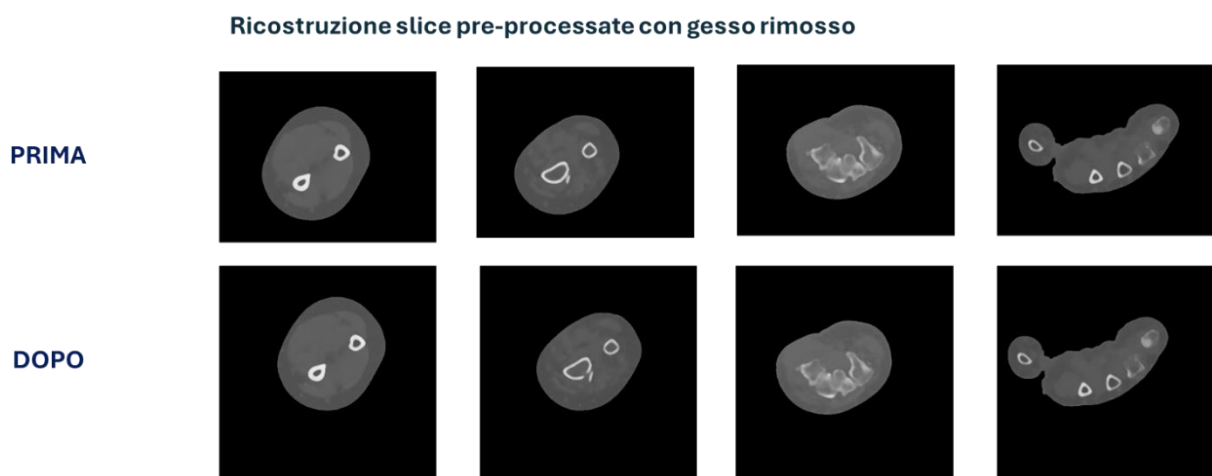


Figura 32: Ricostruzione slice pre-processate con gesso rimosso, prima e dopo la ricostruzione

4.3 Ricostruzione volume

Dopo aver ottenuto le slice ricostruite, il processo è proseguito su MATLAB, dove è stata eseguita la conversione delle slice da formato PNG a formato DICOM utilizzando un codice denominato *PNG2DICOM*, implementato manualmente e descritto dettagliatamente nell'Appendice: Codici. Il codice itera sulle immagini presenti nella cartella contenente i file DICOM originali e le immagini in formato PNG generate dall'algoritmo. Per ogni immagine, estrae i metadati dall'immagine DICOM corrispondente, crea un'istanza di un oggetto DICOM e imposta i metadati appropriati. Successivamente, copia i metadati dall'immagine DICOM di origine all'oggetto DICOM creato e infine scrive l'immagine nel formato DICOM.

Le immagini così salvate vengono importate in 3D SLICER, dove viene applicato un *threshold* per isolare l'osso dal resto del tessuto. I valori soglia sono stati determinati attraverso un processo di tipo “trial and error”, poiché non è stato possibile stabilire un range di soglia universalmente valido tra i diversi pazienti. Questo procedimento ha permesso di ottenere il modello 3D dell'osso.

Dalla visualizzazione 3D è emerso che i modelli più completi e con una morfologia più fedele all'originale sono quelli in cui è stato eseguito l'intero processo di segmentazione. Questo risultato conferma l'ipotesi formulata nel paragrafo "Segmentazione Osso" riguardante l'utilizzo delle maschere dell'osso processate. Pertanto, per procedere con l'allenamento delle reti neurali, si è deciso di utilizzare le maschere dell'osso processate.

In Figura 33 vengono messi a confronto i volumi ottenuti con quello originale. *Process* indica il volume ottenuto a partire dalle segmentazioni effettuate con le maschere dell'osso processate, mentre “*No process*” fa riferimento alle maschere ottenute con l'algoritmo ridotto descritto nel paragrafo “Segmentazione Osso”.



Figura 33: Ricostruzione volume soggetto 41

4.4 Segmentazione con rete neurale

In Figura 34 è mostrato il flowchart completo della segmentazione delle immagini effettuate con l'algoritmo di deep learning.

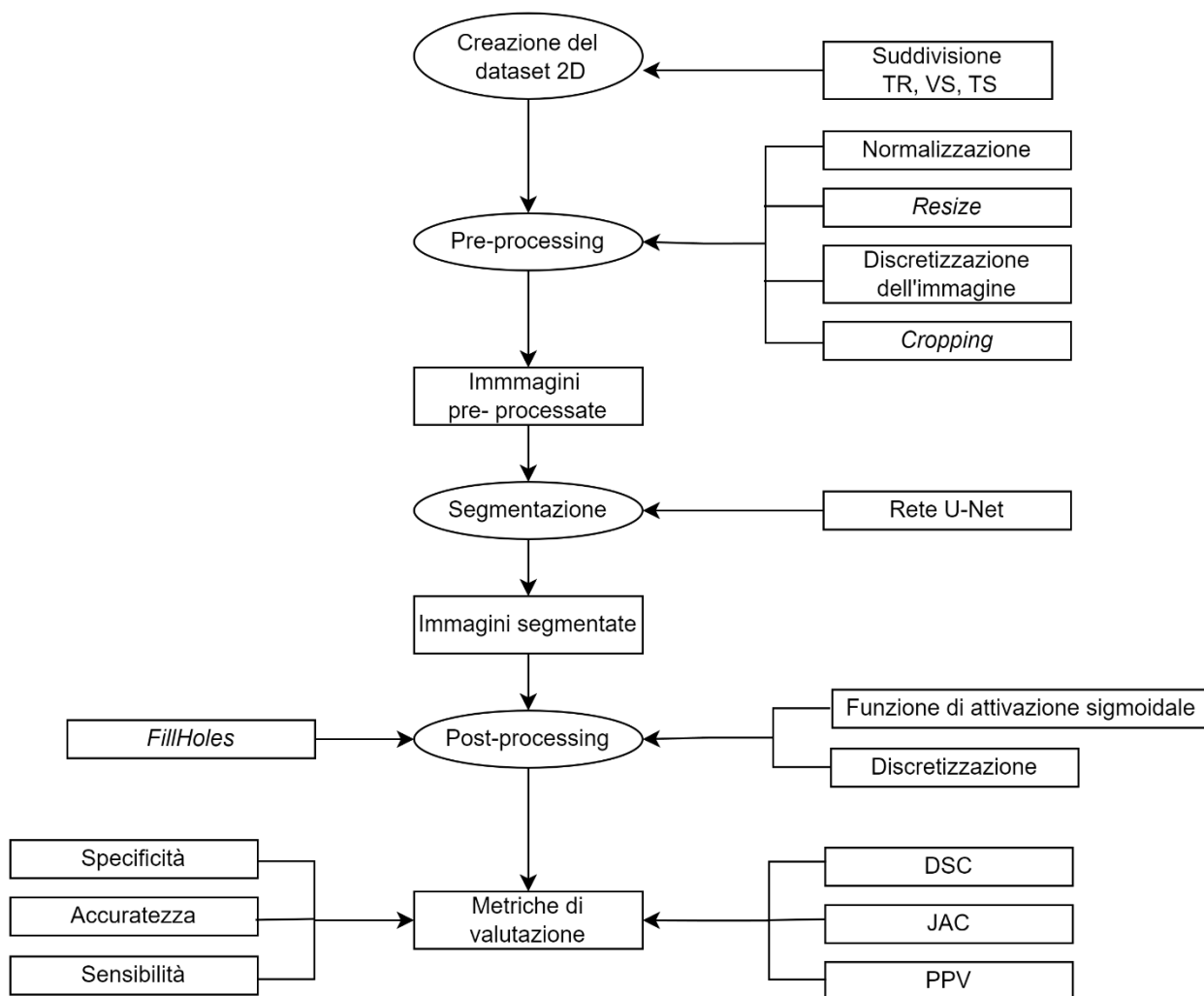


Figura 34: Flowchart Segmentazione con rete neurale

Il processo consta di 6 step principali:

1. Creazione del dataset a partire dalle immagini e maschere ottenute dal processo di segmentazione con l'algoritmo semi-automatico. In questa fase avviene la suddivisione in Training Set (TR), Validation Set (VS) e Test Set (TS);
2. Pre-processing;
3. Segmentazione;
4. Post processing;
5. Calcolo delle metriche;
6. Ricostruzione 3D.

4.4.1 Training set, Validation Set, Test Set

Il dataset utilizzato per lo sviluppo dell'algoritmo di deep learning è composto dalle immagini TC dei pazienti, descritti al paragrafo 4.1. Queste immagini sono state sottoposte a un processo di pre-processing, incluso la rimozione del gesso tramite l'algoritmo di segmentazione manuale sopra descritto.

Per testare l'effetto della rete neurale sia sui singoli pazienti che sull'intero set di test, è stato implementato un codice MATLAB denominato "*Rename_Img*". Questo codice copia le immagini nelle cartelle appropriate e le rinomina nel formato "*PZ_xx_slice_yy*", dove *xx* rappresenta il numero del paziente e *yy* indica il numero della slice. Questa operazione permette di distinguere facilmente i diversi pazienti e di identificare le slice per ciascun paziente.

Il codice è descritto nel dettaglio in Appendice: Codici. Esso è stato progettato per semplificare e uniformare il processo di gestione dei file di immagini e maschere dei pazienti, garantendo che tutti i file siano correttamente nominati e facilmente identificabili.

La suddivisione del dataset in training set, *validation* set e test set è stata effettuata utilizzando un codice Python sviluppato appositamente per questa applicazione, denominato "*Suddivisione_Creazione_Dataset*" e dettagliato in Appendice: Codici.

- **Training set e Validation set**

Il Set di addestramento della rete è composto da 1974 immagini ottenute dalle 15 TC dei pazienti in cui l'algoritmo semi-automatico è stato efficace, ovvero è riuscito ad ottenere una segmentazione corretta delle slice. In queste immagini, il gesso è stato rimosso correttamente.

Nello specifico, il TR rappresenta il 75% delle immagini (14 TC, 1480 immagini), mentre il VS comprende il restante 25% (494 immagini). Per il VS, è stato scelto il paziente con la segmentazione migliore e con il maggior numero di immagini (paziente 46). Di questo soggetto si disponeva della scansione originale. Tale scelta è stata effettuata per testare l'algoritmo su una scansione reale.

Le maschere, invece, sono state ottenute tramite l'algoritmo di CV e successivamente ritoccate manualmente utilizzando l'applicazione MATLAB «ImageLabeler». La quale semplifica il processo di etichettatura delle immagini. Consente agli utenti di evidenziare e assegnare etichette a regioni specifiche delle immagini, migliorando così l'efficienza nel creare set di dati di addestramento accurati per l'addestramento della rete nel caso specifico descritto in questo lavoro.

- **Test set**

Il set per testare la rete è costituito da 907 immagini ottenute dalle 10 TC che l'algoritmo di CV non è stato in grado di segmentare.

Le maschere sono state create manualmente utilizzando 3D Slicer. L'obiettivo è verificare se, addestrando l'algoritmo con le maschere ottenute dall'algoritmo di CV, la rete sia in grado di effettuare una segmentazione il più possibile simile a quella realizzata dall'operatore umano. Poiché solo in 2 dei 10 soggetti il gesso è stato rimosso correttamente, si è deciso di creare due set di test:

1. Test set 1 (Figura 35): le immagini sono state pre-processate dall'algoritmo di CV e con questo è stata applicata la rimozione del gesso. In 8 soggetti, il gesso è ancora presente.

-
2. Test set 2 (Figura 35): la rimozione del gesso è stata eseguita importando su 3D SLICER la serie di immagini pre-processate con l'algoritmo di CV e segmentando la parte interna del gesso, che include osso e muscolo. Successivamente, tramite il codice *Suddivisione_Creazione_Dataset*, è stata effettuata una moltiplicazione tra le immagini processate ricostruite e le maschere appena ottenute.

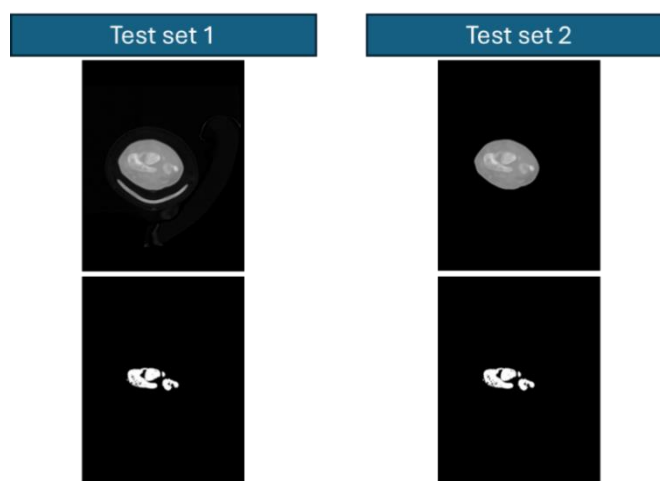


Figura 35: Confronto test set

La distribuzione delle fratture all'interno dei tre set è stata casuale, in quanto la suddivisione dei pazienti si è basata sull'esito dell'algoritmo di segmentazione manuale. Entrando nel dettaglio:

- all'interno del TR sono presenti: 6 fratture di radio distale, 3 fratture dell'epifisi del radio, le 2 fratture non specificate, una frattura dello scafoide-radio sinistro, una frattura dell'epifisi distale, una frattura radio distale e diastasi dello spazio scafo-lunare e una frattura pluriframmentaria scomposta articolare radio distale più frattura composta testa-ulna. La distribuzione è mostrata in
- *Figura 36.*
- Nel VS è stata inserita una frattura del radio distale.
- Il TS è invece composto da 8 fratture del radio distale, una frattura pluriframmentaria scomposta dell'epifisi distale del radio e dell'ulna e una frattura articolare pluriframmentaria scomposta dell'epifisi distale del radio. La distribuzione è mostrata in *Figura 37.*

In tutti i set di dati è presente all'incirca lo stesso numero di fratture distali del radio.

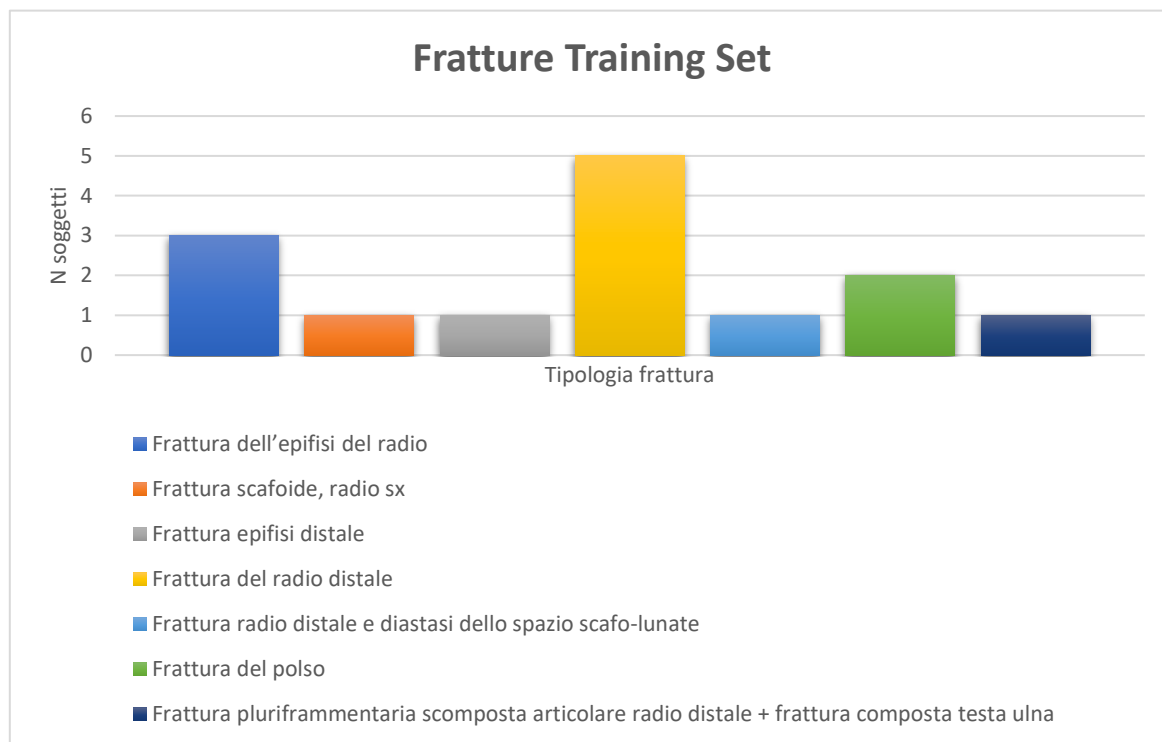


Figura 36: Distribuzione fratture nel training set

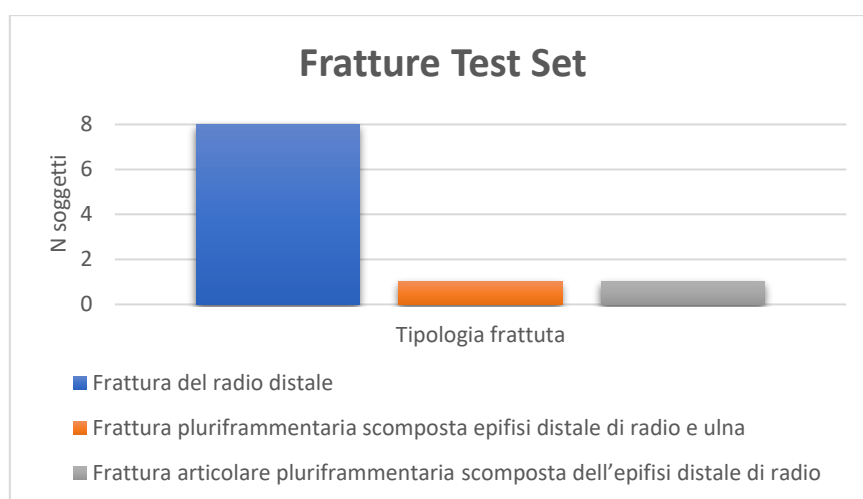


Figura 37: Distribuzione fratture nel testing set

4.4.1.1 Soggetti scartati

Sono stati scartati i soggetti con acquisizioni troppo distanti da quella standard.

Nel caso del soggetto 2, oltre al polso e alla mano di interesse, sono stati acquisiti altri distretti anatomici. Nella Figura 38, sono illustrate due slice diverse. È evidente che all'interno di una stessa slice possono essere presenti entrambe le mani e l'algoritmo di computer vision non è in grado di discriminare quale isolare. Inoltre, nella slice inferiore, sono presenti altre componenti anatomiche oltre alle mani, e il metodo di pre-processing implementato (Data Pre-Processing in paragrafo 3.2) non è sufficiente per eliminarle.



Figura 38: Soggetto 2

Per quanto riguarda il soggetto 9, di cui sono illustrate alcune slice in Figura 39, ci sono componenti di cui non è possibile identificare la natura. Non è chiaro se si tratti di distretti anatomici o elementi dello strumento.



Figura 39: Soggetto 9

Per il soggetto 30, la problematica riscontrata non riguarda l'acquisizione dei dati, ma piuttosto l'estrapolazione. Importando il file DICOM della scansione TC, non è stato possibile estrarre il volume desiderato poiché ogni slice costituisce un file separato. Di conseguenza, non esiste un unico volume che racchiude la vista assiale di interesse. Questo ha impedito l'utilizzo di tale soggetto per lo studio. In Figura 40 b) viene mostrata la schermata di Slicer 3D per il soggetto 30. Come si può notare, a differenza del soggetto 1 in Figura 40 a), selezionando il file viene visualizzata solo una singola slice, come evidente osservando tutte e tre le viste.

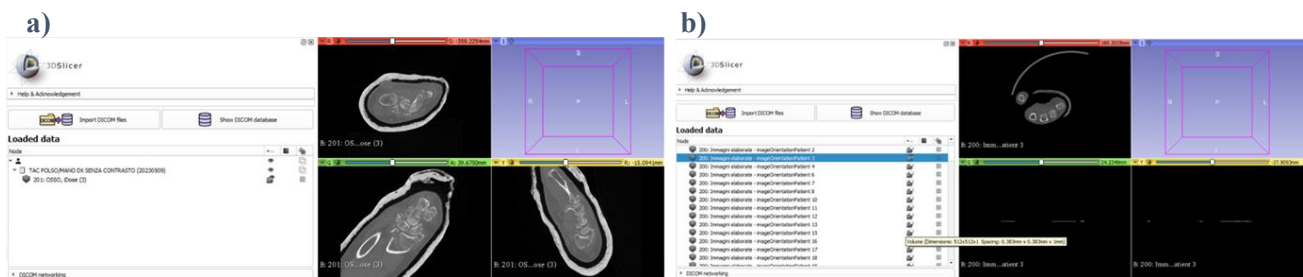


Figura 40: Soggetto 30

4.4.2 Rete neurale

L'algoritmo di segmentazione mediante rete neurale è stato implementato in Google Colab.

La rete più utilizzata per la segmentazione di immagini biomediche è la rete neurale convoluzionale U-Net. (Srinivasan et al., 2024). Questa è la più diffusa anche per quanto riguarda la segmentazione ossea. In particolare, è stata utilizzata per la segmentazione di ulna e radio raggiungendo ottime performance (Gou et al., 2019). Ha ottenuto buoni risultati anche nella segmentazione della mano (Radke et al., 2021).

La rete implementata in questo studio lavora in 2D, in particolare su immagini in scala di grigio, ed è progettata per compiti di segmentazione binaria, dove l'obiettivo è predire la presenza o l'assenza di osso nell'immagine di input.

Per quanto riguarda l'architettura, riportata in Figura 41, U-Net è composta da un encoder e un decoder simmetrici. L'encoder riduce progressivamente la risoluzione spaziale dell'input e aumenta il numero di canali, mentre il decoder ripristina la risoluzione e riduce il numero di canali. Ciò permette alla rete di catturare sia informazioni locali che globali. La sequenza dei canali specifica il numero di filtri nei vari *layer* dell'encoder e del decoder (Noguchi et al., 2020).

Ogni *layer* ha uno stride di 2, il che significa che la dimensione dell'output viene ridotta a metà rispetto all'input. La presenza delle unità residue aiuta a mitigare il problema della riduzione del gradiente durante il *backpropagation* attraverso molteplici *layer* della rete, e questo può migliorare la stabilità e l'efficienza dell'addestramento.

La normalizzazione delle feature avviene mediante la normalizzazione Batch, che è utilizzata per normalizzare le feature all'interno di ciascun *minibatch* durante l'addestramento. Ciò aiuta a garantire una convergenza più rapida e stabile durante l'ottimizzazione.

La funzione di perdita utilizzata è la *Binary Cross-Entropy Loss* (BCELoss), comunemente utilizzata per problemi di classificazione binaria.

Per l'aggiornamento dei pesi della rete durante l'addestramento è stato scelto l'ottimizzatore Adam con un learning rate di 0.001 e un peso di decadimento pari a 0. È efficace nelle applicazioni di deep learning grazie alla sua combinazione di adattività ai gradienti e velocità di convergenza. È computazionalmente efficiente e anche adatto per problemi non stazionari con gradienti molto rumorosi e sparsi (Suganthi et al., 2024).

Il metodo di inferenza utilizzato è il *SimpleInferer*, che applica la rete allenata al singolo batch di input.

La funzione di attivazione utilizzata è di tipo sigmoideale, utilizzata nelle reti neurali per problemi di classificazione binaria, dove si desidera una probabilità di appartenenza a una classe tra 0 e 1 (Noguchi et al., 2020).

Viene inoltre utilizzato un *seed* per rendere deterministico il processo di training e inferenza della rete, ossia per ottenere risultati riproducibili in modo coerente con lo stesso input e configurazione.

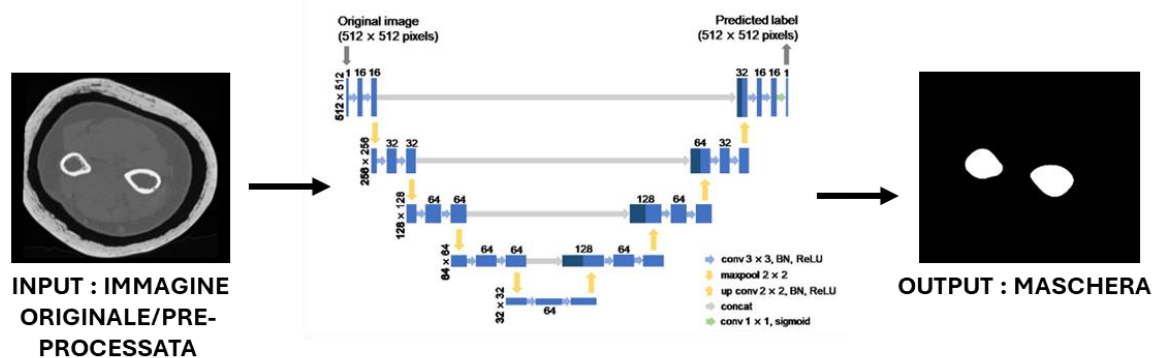


Figura 41: Schema rete per segmentazione U-Net

4.4.2.1 Pre-Processing

Sono stati effettuati alcuni passaggi di pre-processing comuni a tutte le prove utilizzando la funzione *monai.transforms.Compose*, che permette di richiamare le trasformazioni implementate al suo interno in modo sequenziale. Le operazioni di *pre-processing* selezionate sono le seguenti:

- *resize* delle immagini e delle relative segmentazioni a una dimensione spaziale di [512, 512], per garantire che le dimensioni delle immagini del dataset siano uniformi;
- normalizzazione dei valori dei pixel dell'immagine e della segmentazione da due range di valori rispettivamente pari a [0, 255] e [0, 1];
- discretizzazione della maschera manuale, convertendo i valori dei pixel della segmentazione in valori binari (0 o 1) applicando una soglia di 0.5;
- *crop* delle immagini con dimensioni 400x400. (Hendrix et al., 2021)

4.4.2.2 Post-Processing

Analogamente al *pre-processing*, la fase di post-processing è stata effettuata tramite la funzione *monai.transforms.Compose*, applicando operazioni comuni a tutte le prove. Le trasformazioni applicate sono le seguenti:

- Applicazione di una funzione di attivazione di tipo sigmoideale che comprime i valori in un range tra 0 e 1. Questa operazione è necessaria perché il modello utilizzato nella fase di predizione restituisce valori continui anziché una segmentazione binaria.
- Discretizzazione della maschera tramite l'applicazione di una soglia di 0.5 per convertire i valori della segmentazione in 0 o 1 ottenendo una segmentazione binaria.

4.4.2.3 Data Augmentation

Poiché il dataset disponibile è limitato, si utilizza la *Data Augmentation* per ampliare il dataset e migliorare la robustezza e la generalizzazione dei modelli di elaborazione delle immagini mediche.

Sono state generate nuove immagini a partire da quelle iniziali con una probabilità del 50% attraverso le seguenti tecniche:

- una rotazione di una quantità compresa tra -20° e 20° , sia rispetto all'asse orizzontale che a quello verticale (Radke et al., 2021);
- un *flipping* rispetto all'asse orizzontale (Radke et al., 2021).

Nello specifico il TR di partenza è composto da 1480 immagini. Applicando la *Data Augumentation* si aggiunge un 50% di immagini, portando il TR finale a circa 2250 immagini.

Queste trasformazioni hanno come obiettivo l'aumentare la varietà dei dati di addestramento, consentendo ai modelli di apprendere da una gamma più ampia di condizioni e angolazioni, migliorando così le prestazioni e la capacità di generalizzazione del modello.

4.4.2.4 Allenamento rete

Per l'allenamento della rete sono stati utilizzati batch di 2 immagini. Dal momento che la morfologia dell'osso del polso può essere molto differente tra una slice e l'altra, un batch size molto basso assicura un miglior apprendimento della rete.

Il numero di iterazioni per effettuare la validazione di ogni epoca è stato preso pari alla potenza di 2 più vicina al rapporto tra le dimensioni del TR e le dimensioni del batch (512). Il massimo numero di batch per completare un'epoca è stato considerato pari a 10 volte il numero di iterazioni.

Il modello di addestramento utilizza il metodo del *transfer learning* in quanto riduce significativamente i tempi e non ha bisogno di un enorme set di dati. Nella sua forma più generale, consiste nelle seguenti quattro fasi principali (Öztürk et al., 2023):

1. Recupero dei livelli del modello pre-addestrato.
2. Congelamento degli *layer* per evitare la perdita di informazioni nei futuri cicli di addestramento.
3. Aggiunta di nuovi *layer* addestrabili in cima al modello congelato.
4. Training di nuovi *layer*.

Per la validazione della rete si fa inferenza, il che significa che il modello viene utilizzato solo per fare previsioni e non per aggiornare i suoi pesi come avviene durante l'addestramento. Si itera attraverso le immagini del set di validazione e le relative maschere che indicano la posizione degli oggetti all'interno delle immagini. Utilizzando il modello precedentemente addestrato, vengono fatte previsioni sulle immagini di input. Queste previsioni sono confrontate con le maschere di validazione per calcolare la perdita (*loss*), che rappresenta la discrepanza tra le previsioni del modello e le maschere reali.

Le immagini di input, le maschere di output predette dal modello e le maschere reali vengono salvate.

4.4.3 Prove effettuate

Sono state addestrate quattro reti neurali differenti, suddivise in due gruppi principali:

1. Reti senza applicazione della Data Augmentation (Figura 42 a)
 - a. Allenamento senza post-processing;
 - b. Allenamento con post-processing fornito dalla libreria Monai.
2. Reti con applicazione del Data Augmentation (Figura 42 b):
 - a. Allenamento senza post-processing;
 - b. Allenamento con post-processing fornito dalla libreria Monai.

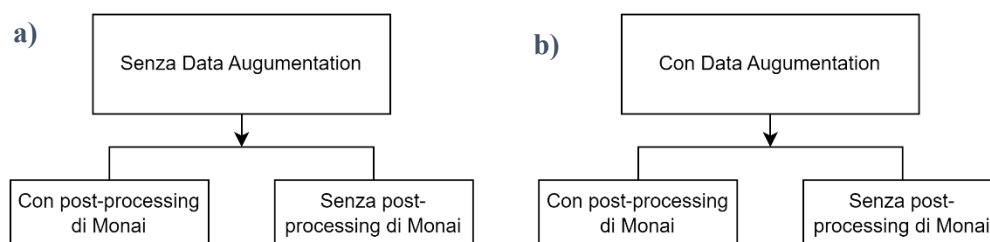


Figura 42: Addestramenti rete neurale

Per quanto riguarda le prove di test, sono stati utilizzati due approcci differenti:

1. Test su un dataset costituito da tutte le immagini dei pazienti di test (Figura 43a): in questo caso, sono state utilizzate sia le immagini in cui il gesso è stato rimosso tramite l'algoritmo di segmentazione semi-automatica, sia quelle in cui la rimozione è stata effettuata manualmente dall'operatore.

Per entrambe le tipologie di immagini, sono state condotte prove con e senza *Data Augmentation*, e con e senza l'applicazione del post-processing della libreria Monai, in accordo con le reti neurali addestrate.

2. Test utilizzando ogni singolo paziente come dataset (Figura 43b): anche in questo caso, sono state eseguite le stesse prove su immagini dei singoli soggetti, includendo le varianti con e senza *Data Augmentation* e con e senza post-processing di Monai.

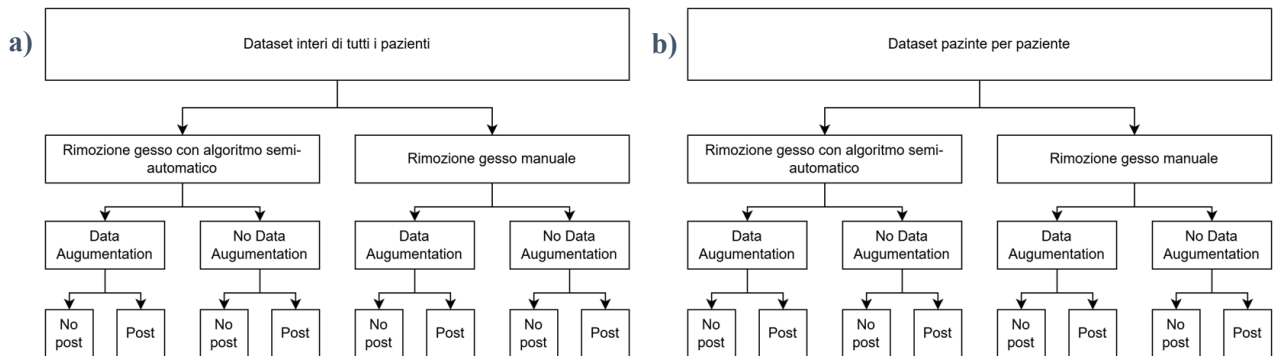


Figura 43: Prove test set

L'obiettivo della differenziazione tra i due approcci è valutare l'efficienza dell'algoritmo di segmentazione semi-automatica nella rimozione del gesso e analizzare come variano le performance delle reti in base alle diverse problematiche legate alle immagini dei singoli pazienti.

4.4.4 Metriche

Per valutare le prestazioni della rete sono state confrontate le singole slice della maschera manuale (MM) e di quella predetta (PM). In Figura 44 vengono mostrati due esempi di maschera manuale e maschera predetta.



Figura 44: Esempio maschera manuale e maschera predetta

Di seguito sono descritte le metriche utilizzate per la valutazione delle prestazioni:

- Il *Dice Similarity Coefficient* (DSC), che rappresenta la sovrapposizione tra la maschera predetta e quella manuale (Julia Mai, 2024). È definito come 2 volte l'intersezione tra MM e PM diviso per la somma di MM e PM:

$$\text{DSC} = 2 * \frac{|MM \cap PM|}{|MM| + |PM|} \quad (\text{Shen et al., 2023})$$

Il suo valore è compreso tra 0 e 1, in cui 0 indica che le due maschere non hanno pixel in comune e 1 indica che sono completamente sovrapposte.

- *Jaccard index* (JAC), definito come il rapporto tra intersezione e unione delle due maschere, valuta la somiglianza tra MM e PM ed è compreso in range di valori da 0 a 1. Se JAC è pari a 0 le due maschere sono completamente diverse mentre se è uguale le due maschere sono uguali.

$$\text{JAC} = \frac{|MM \cap PM|}{|MM \cup PM|} \quad (\text{Eelbode et al., 2020})$$

Queste due metriche vengono utilizzate perché sono l'indicatore più adeguato alla qualità percettiva di una segmentazione. (Eelbode et al., 2020)

- Sensitività, indica la percentuale di pixel correttamente classificati come appartenenti o non appartenenti alla regione di interesse rispetto al totale dei pixel esaminati.

$$\text{Sens} = \frac{TP}{TP + FN} \quad (\text{Dalal et al., 2024})$$

- Accuratezza, misura la capacità del modello di identificare correttamente i pixel che appartengono alla regione di interesse.

$$\text{Acc} = \frac{TP + TN}{TP + FP + TN + FN} \quad (\text{Dalal et al., 2024})$$

- Specificità, rappresenta la capacità del modello di identificare correttamente i pixel che non appartengono alla regione di interesse.

$$\text{Spec} = \frac{TN}{FP + TN} \quad (\text{Dalal et al., 2024})$$

-
- PPV, indica la probabilità che i pixel classificati come appartenenti alla regione di interesse siano effettivamente tali.

$$PPV = \frac{TP}{FP + TP} \text{ (Dalal et al., 2024)}$$

Alla fine del processo di validazione su tutti i batch, vengono calcolate le medie e le deviazioni standard di tutti i parametri. Tali valori rappresentano una stima complessiva delle prestazioni del modello sul set di dati di validazione

4.4.5 Ricostruzione 3D

Dopo la rimozione semi-automatica del gesso e l'estrazione delle maschere tramite rete neurale, è stata isolata la porzione ossea di interesse. Successivamente, le slice ricostruite sono state elaborate in MATLAB, dove è stata effettuata la conversione delle immagini da formato PNG a DICOM tramite il codice *PNG2DICOM*, implementato manualmente e descritto dettagliatamente nell'Appendice: Codici. Questo script itera sulle immagini presenti nella cartella contenente i file DICOM originali e le corrispondenti immagini PNG generate dalla rete neurale. Per ogni immagine, il codice estrae i metadati dal file DICOM di origine, crea un nuovo oggetto DICOM, assegna i metadati appropriati e infine salva l'immagine convertita nel formato DICOM.

Le immagini così ottenute sono state importate in 3D SLICER, dove è stato applicato un *threshold* per separare l'osso dal resto dei tessuti. La soglia di segmentazione è stata determinata tramite un approccio "trial and error", poiché non è stato possibile individuare un valore universale valido per tutti i pazienti. A questo punto, è stato eseguito un post-processing per rimuovere eventuali pixel spuri erroneamente identificati come tessuto osseo a causa di intensità simili. Questo passaggio ha permesso di ottenere un modello 3D dell'osso.

Dalla visualizzazione tridimensionale è emerso che il modello non risulta perfettamente preciso e non consente un'identificazione chiara e netta delle singole componenti ossee del polso. Tuttavia, l'anatomia riprodotta è complessivamente fedele a quella reale. In Figura 45 sono messi a confronto due modelli 3D del polso: (b) quello ottenuto con le segmentazioni della rete neurale e (a) quello generato dalle immagini originali. Si osserva che, nel modello basato sulla rete neurale, la frattura non è immediatamente evidente, ma

confrontandolo con il modello di riferimento si nota che essa è comunque ricostruita, pur con i limiti sopra descritti.

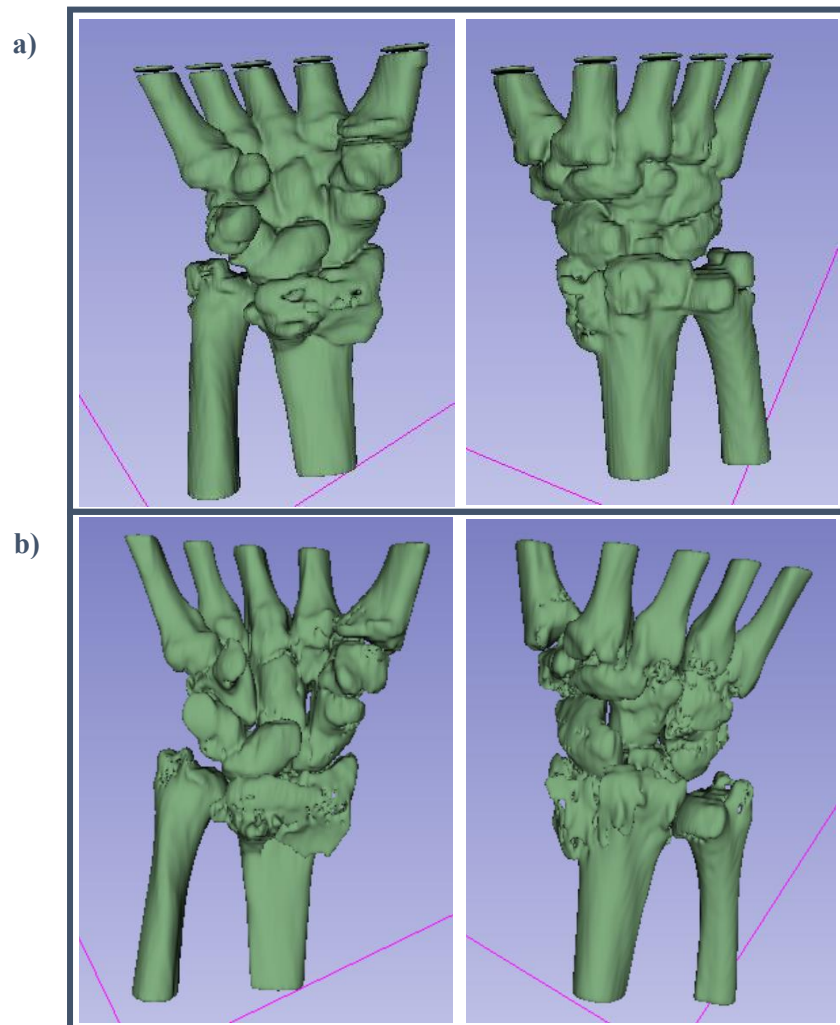


Figura 45: Confronto ricostruzione 3D del polso ottenuta dalle immagini originali (a) con le segmentazioni generate dalla rete neurale (b).

5 Risultati

TRAINING SET E VALIDATION SET

Analizzando la Tabella 1 e la Tabella 2, emerge che l'uso della Data Augmentation senza post-processing porta a un significativo calo nel JAC, nella Sensibilità, e nel PPV, indicando difficoltà nel rilevare correttamente i casi positivi. Tuttavia, l'introduzione del post-processing migliora leggermente lo Jaccard Index, e porta a un aumento significativo della Sensibilità e del PPV. Questo suggerisce che il post-processing è efficace nel mitigare alcuni dei problemi introdotti dal data augmentation.

Nei modelli addestrati senza Data Augmentation, l'applicazione del post-processing non evidenzia miglioramenti significativi nelle performance.

Complessivamente, il modello ottiene le prestazioni migliori (DSC (0.92 ± 0.02) e accuratezza (99.29 ± 0.7) più elevate,) quando non viene utilizzata la Data Augmentation. Si è notato che il post-processing non altera significativamente le performance. La specificità rimane costantemente alta in tutte le configurazioni, indicando una buona capacità del modello di identificare correttamente i negativi.

Tabella 1: Risultati ottenuti sul training set

Training set							
		DSC	JAC (%)	Accuratezza (%)	Specificità (%)	Sensibilità (%)	PPV (%)
No Data Augmentation	No Post-processing	0.92 ± 0.02	76.52 ± 19.93	99.29 ± 0.67	99.55 ± 0.51	88.12 ± 19.96	83.08 ± 18.82
	Post-processing	0.92 ± 0.02	76.4 ± 20.22	99.27 ± 0.78	99.5 ± 0.68	88.75 ± 20.02	82.46 ± 19.24
	No Post-processing	0.90 ± 0.02	42.59 ± 30.98	97.65 ± 1.94	98.68 ± 1.11	55.51 ± 35.59	51.84 ± 33.16
	Post-processing	0.92 ± 0.02	44.51 ± 32.77	99.33 ± 0.64	99.58 ± 0.50	88.73 ± 19.27	97.29 ± 0.0

Tabella 2: Risultati ottenuti sul Validation set

Validation set							
		DSC	JAC (%)	Accuratezza (%)	Specificità (%)	Sensibilità (%)	PPV (%)
No Data Augmentation	No Post-processing	0.92± 0.02	76.52 ± 19.93	99.3 ± 1.38	99.45 ± 1.36	95.42 ± 19.96	87.75 ± 16.37
	Post-processing	0.92± 0.02	85.09 ± 16.38	99.32 ± 1.53	99.42 ± 1.51	96.12 ± 6.59	87.74 ± 16.63
	No Post-processing	0.90 ± 0.02	42.59 ± 30.98	97.44 ± 2.13	98.52 ± 1.40	58.34 ± 35.52	51.84 ± 33.16
	Post-processing	0.92± 0.02	44.51 ± 32.77	99.33 ± 1.85	99.38 ± 1.84	96.86 ± 5.62	91.93 ± 0.0

TESTING SET OTTENUTO CON ALGORITMO SEMI-AUTOMATICO

Dataset Completo

Analizzando i risultati ottenuti con il dataset completo (Tabella 3), dal quale il gesso è stato rimosso tramite un algoritmo semi-automatico, emerge che senza Data Augmentation e senza Post Processing il DSC medio è 0.51 ± 0.02 e lo Jaccard Index è 35.44 ± 15.28 . Questo indica una performance accettabile, ma non ottimale. Aggiungendo il post-processing, le performance non migliorano, anzi il DSC medio scende leggermente a 0.47 ± 0.03 e lo Jaccard Index a 33.23 ± 15.91 , suggerendo che il post-processing potrebbe introdurre rumore o artefatti.

Nei test effettuati con Data Augmentation senza Post Processing, il DSC medio migliora a 0.54 ± 0.02 e lo Jaccard Index a 36.82 ± 15.56 . Questo suggerisce che la Data Augmentation migliora la performance complessiva della rete. Tuttavia, la sensibilità media è relativamente bassa ($62.86 \pm 35.57\%$).

Con il Post Processing, il DSC medio scende a 0.43 ± 0.06 e lo Jaccard Index a 30.6 ± 16.41 , suggerendo che il post-processing potrebbe non essere utile quando si utilizza la Data Augmentation.

Nonostante tutto, la configurazione ottimale risulta essere la combinazione di Data Augmentation e post-processing, poiché offre i risultati globali migliori, con metriche più bilanciate e alte per accuratezza, specificità, sensibilità e anche il PPV più elevato (Tabella 4, Tabella 5, Tabella 6).

Variazioni tra i pazienti

Vi sono variazioni significative nelle performance per diversi pazienti, suggerendo che ulteriori ottimizzazioni potrebbero beneficiare la rete neurale per casi specifici.

Nelle prove senza Data Augmentation e Post Processing (Tabella 3), i valori di DSC variano da 0.32 a 0.69, con la maggior parte dei valori intorno a 0.4-0.5, mentre lo Jaccard Index varia dal 25.79% al 52.43%, con deviazioni standard che indicano una variabilità moderata. Dalla Tabella 4 si evince che l'accuratezza e la specificità presentano valori mediamente elevati, mentre la sensibilità varia considerevolmente, con una media più bassa rispetto alla specificità (da 71.58% a 99.71%). I valori di PPV si trovano in un range da 16.33% a 54.16% (Tabella 6), indicando una bassa probabilità di veri positivi tra i positivi predetti.

Aggiungendo il Post Processing (Tabella 3), il DSC peggiora nella maggior parte dei pazienti rispetto allo scenario senza post-processing, mentre lo Jaccard Index peggiora, con deviazioni standard moderate. L'accuratezza migliora leggermente, con valori che variano da 91.78% a 97.86%. Specificità, sensibilità e PPV sono in linea con il primo scenario (Tabella 4, Tabella 6).

Nelle prove effettuate con Data Augmentation, i valori di DSC e Jaccard Index migliorano, così come accuratezza, PPV e specificità, mentre la sensibilità peggiora (Tabella 3, Tabella 5, Tabella 6). L'aggiunta del post-processing peggiora le performance nella maggior parte dei pazienti.

È importante notare che i pazienti 22 e 32 mostrano le migliori performance globali, mantenendo alti valori nelle metriche in tutti gli scenari. Infatti, questi sono gli unici in cui l'algoritmo è riuscito a rimuovere il gesso correttamente.

Tabella 3: Risultati (DSC e JAC) ottenuti con il Test set 1

Testing set: immagini ottenute con algoritmo semi-automatico								
N° Pazienti	No Data Augumentation				Data Augumentation			
	No Post-processing		Post-processing		No Post-processing		Post-processing	
	DSC	JAC(%)	DSC	JAC(%)	DSC	JAC(%)	DSC	JAC(%)
14	0.44 ± 0.02	25.79 ± 9.03	0.37 ± 0.05	18.84 ± 5.27	0.52 ± 0.03	32.28 ± 8.95	0.31 ± 0.03	15.29 ± 4.37
15	0.6 ± 0.03	37.08 ± 10.59	0.55 ± 0.04	30.05 ± 10.30	0.60 ± 0.01	38.49 ± 10.57	0.49 ± 0.03	27.49 ± 6.94
16	0.46 ± 0.05	21.33 ± 10.57	0.37 ± 0.04	17.74 ± 9.72	0.51 ± 0.05	24.49 ± 11.1	0.35 ± 0.05	14.51 ± 8.25
22	0.66 ± 0.01	42.8 ± 8.71	0.65 ± 0.03	41.0 ± 10.19	0.65 ± 0.01	41.79 ± 7.84	0.66 ± 0.02	41.2 ± 12.63
25	0.55 ± 0.05	38.43 ± 9.18	0.51 ± 0.07	35.43 ± 10.47	0.56 ± 0.05	39.07 ± 9.43	0.47 ± 0.05	28.93 ± 8.62
27	0.47 ± 0.05	38.45 ± 14.53	0.51 ± 0.04	36.63 ± 14.73	0.51 ± 0.07	41.71 ± 15.68	0.47 ± 0.07	35.97 ± 13.73
28	0.53 ± 0.05	36.01 ± 7.01	0.50 ± 0.05	33.93 ± 5.87	0.53 ± 0.04	33.1 ± 8.73	0.45 ± 0.02	29.07 ± 5.88
29	0.47 ± 0.05	27.12 ± 14.45	0.38 ± 0.04	21.93 ± 11.89	0.52 ± 0.06	30.48 ± 16.15	0.30 ± 0.03	15.63 ± 8.32
31	0.69 ± 0.01	16.26 ± 13.47	0.37 ± 0.11	15.75 ± 14.21	0.36 ± 0.06	18.63 ± 14.11	0.33 ± 0.06	15.86 ± 14.1
32	0.32 ± 0.03	52.43 ± 11.83	0.69 ± 0.01	41.61 ± 13.22	0.70 ± 0.01	54.32 ± 11.49	0.7 ± 0.01	54.07 ± 13.55
Media tutti i pazienti	0.51 ± 0.02	35.44 ± 15.28	0.47 ± 0.03	33.23 ± 15.91	0.54 ± 0.02	36.82 ± 15.56	0.43 ± 0.04	30.6 ± 16.41

Tabella 4: Risultati (Accuratezza, Specificità, Sensibilità) ottenuti senza Data Augmentation

Testing set: immagini ottenute con algoritmo semi-automatico						
N° Pazienti	No Data Augmentation					
	No Post-processing			Post-processing		
	Accuratezza (%)	Specificità (%)	Sensibilità (%)	Accuratezza (%)	Specificità (%)	Sensibilità (%)
14	95.5 ± 3.18	95.44 ± 3.23	99.71 ± 0.65	93.15 ± 2.13	94.16 ± 1.98	14.17 ± 12.21
15	94.8 ± 4.51	94.69 ± 4.74	96.61 ± 14.5	92.87 ± 6.50	92.64 ± 6.86	97.7 ± 14.41
16	93.62 ± 4.33	93.54 ± 4.41	98.44 ± 11.09	91.78 ± 5.41	91.67 ± 5.51	98.71 ± 10.31
22	97.98 ± 0.65	98.01 ± 0.64	94.75 ± 9.3	97.68 ± 1.08	97.69 ± 1.07	95.66 ± 8.97
25	96.89 ± 9.56	96.92 ± 9.57	93.98 ± 14.73	96.26 ± 9.68	96.25 ± 9.70	95.36 ± 14.35
27	98.9 ± 0.57	98.94 ± 0.62	89.31 ± 28.01	98.72 ± 0.86	98.76 ± 0.90	89.22 ± 28.07
28	98.35 ± 0.58	98.37 ± 0.64	95.16 ± 14.45	98.15 ± 0.75	98.16 ± 0.80	95.41 ± 14.51
29	98.02 ± 1.02	98.32 ± 1.12	79.51 ± 39.79	97.29 ± 1.52	97.58 ± 1.62	79.83 ± 39.92
31	95.63 ± 2.74	95.67 ± 2.75	88.13 ± 19.5	94.55 ± 4.06	94.56 ± 4.10	89.8 ± 20.01
32	98.0 ± 1.67	98.11 ± 1.70	94.88 ± 11.04	97.86 ± 2.95	97.89 ± 2.99	96.95 ± 10.2
Media tutti i pazienti	97.27 ± 4.18	97.31 ± 4.24	92.64 ± 20.32	96.65 ± 4.89	96.66 ± 4.97	93.41 ± 20.31

Tabella 5: Risultati (Accuratezza, Specificità, Sensibilità) ottenuti con Data Augmentation

Testing set: immagini ottenute con algoritmo semi-automatico						
N° Pazienti	Data Augmentation					
	No Post-processing			Post-processing		
	Accuratezza (%)	Specificità (%)	Sensibilità (%)	Accuratezza (%)	Specificità (%)	Sensibilità (%)
14	96.14 ± 2.37	96.47 ± 2.27	55.99± 39.98	92.53 ± 3.51	92.42 ± 3.58	99.93± 0.32
15	94.1 ± 4.01	94.46 ± 4.32	67.39± 32.07	91.38 ± 7.05	91.1 ± 7.45	97.75± 14.42
16	93.92 ± 3.94	94.13 ± 4.06	76.21± 29.9	90.32 ± 4.93	90.18 ± 5.02	98.7 ± 10.9
22	97.11 ± 1.09	97.49 ± 0.88	68.95± 29.97	97.58± 1.27	97.6± 1.25	95.65 ± 9.03
25	96.03 ± 9.54	96.45 ± 9.54	64.66± 34.09	95.31 ± 9.74	95.26 ± 9.77	96.3 ± 13.99
27	98.67 ± 0.72	98.89 ± 0.65	65.11± 33.86	98.66 ± 0.88	98.69 ± 0.94	90.5± 28.23
28	97.45 ± 0.98	97.83 ± 0.74	54.32± 41.23	97.43 ± 1.46	97.44 ± 1.52	95.47± 14.52
29	98.08 ± 0.76	98.53 ± 0.98	58.78 ±41.81	96.1 ± 2.05	96.4 ± 2.28	79.95± 39.98
31	95.81± 2.68	96.03 ± 2.63	55.22± 37.79	94.41 ± 3.98	94.4 ± 4.02	93.67± 16.47
32	97.4 ±10.2	97.85 ± 1.34	73.01± 28.83	97.55 ± 4.57	97.54 ± 4.64	98.21 ± 9.47
Media tutti i pazienti	96.83 ± 4.05	97.18 ± 4.06	62.6± 35.57	94.4 ± 4.02	97.54 ± 4.02	93.54± 19.99

Tabella 6: PPV ottenuto sul Test set 1

Testing set: immagini ottenute con algoritmo semi-automatico				
N° Pazienti	No Data Augumentation		Data Augumentation	
	No Post-processing	Post-processing	No Post-processing	Post-processing
	PPV (%)	PPV (%)	PPV (%)	PPV (%)
14	25.82± 9.05	19.96 ± 6.27	18.92 ± 16.59	11.92 ± 3.91
15	37.3 ± 10.67	31.25 ± 9.89	25.55 ± 13.84	32.27 ± 8.03
16	21.38 ± 10.56	17.77 ± 9.69	18.61 ± 11.72	23.24 ± 8.22
22	43.68 ± 8.18	41.62 ± 9.87	31.34 ± 15.13	46.53 ± 12.42
25	39.32 ± 9.7	35.98 ± 10.85	26.59 ± 14.98	21.26 ± 8.86
27	38.73 ± 14.59	36.84 ± 14.72	30.9 ± 17.88	43.73 ± 13.78
28	36.49 ± 6.71	34.35 ± 5.63	19.65 ± 15.85	30.83 ± 5.92
29	27.21 ± 14.55	21.95 ± 11.9	22.55 ± 17.03	18.38± 8.32
31	16.33 ± 13.43	15.78 ± 14.19	10.59 ± 12.68	37.67 ± 14.13
32	54.16 ± 12.8	54.18 ± 13.27	43.37 ± 19.92	45.12 ± 13.75
Media tutti i pazienti	35.99 ± 15.69	33.57± 16.13	25.28 ± 17.98	37.25 ± 15.81

TESTING SET OTTENUTO CON RIMOZIONE MANUALE DEL GESSO

Dataset completo

Senza Data Augumentation e post-processing, il DSC è piuttosto basso (0,61). L'introduzione del post-processing non cambia significativamente il DSC, così come l'introduzione della Data Augumentation senza post-processing non ha un impatto significativo. L'uso combinato di entrambe le tecniche fornisce un piccolo incremento nel DSC (Tabella 7).

Anche nello JAC si osserva un leggero miglioramento con l'uso combinato di Data Augumentation e post-processing. Tuttavia, l'impatto complessivo di entrambe le tecniche è minimo.

L'accuratezza rimane elevata in tutte le configurazioni, con valori molto simili tra loro (circa 98%). Questo suggerisce che la rete U-Net è generalmente affidabile nel distinguere tra osso e non osso. (Tabella 8 e Tabella 9)

La specificità è anch'essa alta e costante, attorno all'88%, indicando una buona capacità del modello di identificare correttamente i veri negativi (Tabella 8 e Tabella 9).

La sensibilità mostra una maggiore variabilità rispetto ad accuratezza e specificità. Il valore più alto di sensibilità si ottiene con entrambe le tecniche di Data Augumentation e post-processing (sensibilità pari a 92.95 ± 20.47 %), suggerendo che queste tecniche aiutano la rete a migliorare nel riconoscimento corretto dei veri positivi (Tabella 8 e Tabella 9).

Il PPV misura la proporzione di veri positivi tra tutti i positivi predetti quindi un valore più alto indica una migliore precisione nella predizione delle regioni ossee. Senza Data Augumentation e post-processing, il PPV è piuttosto basso con elevata variabilità. L'aggiunta del post-processing lo migliora leggermente, suggerendo che aiuta a ridurre i falsi positivi. La Data Augumentation senza post-processing offre, invece, un miglioramento marginale. (Tabella 10)

L'uso combinato Data Augumentation e post-processing fornisce i migliori risultati in termini di PPV, suggerendo che insieme migliorano significativamente la precisione del modello nel predire correttamente le regioni ossee.

In conclusione, la combinazione di Data Augmentation e post-processing fornisce i migliori risultati in termini di sensibilità e PPV, senza compromettere l'accuratezza e la specificità. Questo indica che tali tecniche aiutano il modello a generalizzare meglio, migliorando la capacità di segmentazione ossea nelle immagini TC.

In generale, i miglioramenti apportati da Data Augmentation e post-processing sono presenti ma non sostanziali. Questo suggerisce che potrebbero essere necessari ulteriori affinamenti del modello o l'esplorazione di altre tecniche di pre-processing o post-processing per ottenere miglioramenti più significativi.

Variazioni tra i pazienti

La Data Augmentation tende a migliorare le performance medie sia del DSC che dello Jaccard Index per alcuni pazienti. Tuttavia, in alcuni casi, come il Paziente 27, il post-processing può non essere benefico sia utilizzato con Data Augmentation sia senza. In assenza di Data Augmentation il post-processing può migliorare le metriche ma in alcuni casi, può anche peggiorarle (Tabella 7).

In generale, Data Augmentation e post-processing hanno effetti variabili sull'accuratezza. Alcuni pazienti (es. Paziente 14 e 15) mostrano miglioramenti con Data Augmentation e post-processing, mentre altri (es. Paziente 32) mostrano variazioni significative (Tabella 9).

La specificità tende a essere elevata per tutti i pazienti in tutte le configurazioni, con minime variazioni. Data Augmentation e post-processing non mostrano un impatto significativo (Tabella 8 e Tabella 9).

La sensibilità mostra una maggiore variabilità rispetto ad accuratezza e specificità. In particolare, pazienti come il Paziente 29 e 31 mostrano notevoli miglioramenti con Data Augmentation e post-processing. (Tabella 8 e Tabella 9)

Il post-processing tende a migliorare il PPV per la maggior parte dei pazienti, specialmente in assenza di Data Augmentation. Tuttavia, per alcuni pazienti (es. 15, 28, 32), il post-processing può ridurre il PPV, suggerendo che l'efficacia del post-processing può variare significativamente a seconda del paziente e della configurazione utilizzata (Tabella 10).

Pazienti come il Paziente 14 mostrano miglioramenti in tutte le metriche con l'uso combinato di Data Augmentation e post-processing. Al contrario, il Paziente 32 mostra una riduzione significativa della sensibilità con Data Augmentation e post-processing.

Vi sono, quindi, variazioni significative tra pazienti. Non esiste una configurazione unica ottimale per tutti i pazienti. Ogni paziente risponde in modo diverso alle combinazioni di Data Augmentation e post-processing.

Tuttavia, la combinazione di Data Augmentation e post-processing tende a migliorare le performance medie della maggior parte dei pazienti, suggerendo la necessità di approcci personalizzati per l'ottimizzazione.

Il paziente 32 anche in questo caso presenta le performance migliori.

Risultati Test Set 2: DSC, JAC								
N° Pazienti	No Data Augumentation				Data Augumentation			
	No Post-processing		Post-processing		No Post-processing		Post-processing	
	DSC	JAC(%)	DSC	JAC(%)	DSC	JAC(%)	DSC	JAC(%)
14	0.66 ± 0.04	25.79 ± 9.03	0.63 ± 0.04	40.41 ± 11.6	0.61 ± 0.03	39.45 ± 8.37	0.67 ± 0.03	43.56 ± 10.55
15	0.7 ± 0.006	51.66 ± 14.39	0.69 ± 0.01	49.63 ± 14.53	0.7 ± 0.01	50.42 ± 13.72	0.7 ± 0.01	52.27 ± 13.97
16	0.59 ± 0.03	35.18 ± 8.22	0.56 ± 0.04	32.6 ± 9.92	0.56 ± 0.03	32.96 ± 8.95	0.6 ± 0.03	35.47 ± 9.5
22	0.66 ± 0.02	43.66 ± 7.48	0.65 ± 0.03	41.0 ± 10.19	0.65 ± 0.01	42.73 ± 6.71	0.67 ± 0.02	45.12 ± 7.76
25	0.58 ± 0.05	42.27 ± 9.49	0.58 ± 0.07	41.54 ± 9.84	0.59 ± 0.06	42.7 ± 6.23	0.60 ± 0.06	43.59 ± 9.61
27	0.5 ± 0.08	43.19 ± 15.8	0.49 ± 0.09	43.66 ± 15.98	0.62 ± 0.03	46.69 ± 16.02	0.49 ± 0.09	44.29 ± 16.1
28	0.55 ± 0.05	36.85 ± 7.72	0.54 ± 0.05	36.92 ± 7.92	0.55 ± 0.05	36.88 ± 7.22	0.56 ± 0.02	37.19 ± 7.87
29	0.54 ± 0.05	32.21 ± 17.47	0.53 ± 0.05	31.65 ± 17.17	0.55 ± 0.06	32.25 ± 17.08	0.4 ± 0.1	32.88 ± 17.87
31	0.51 ± 0.01	33.16 ± 7.21	0.52 ± 0.01	33.74 ± 7.12	0.54 ± 0.01	36.27 ± 6.78	0.52 ± 0.01	34.46 ± 6.70
32	0.69 ± 0.02	53.55 ± 93.14	0.69 ± 0.02	54.48 ± 9.97	0.7 ± 0.02	55.34 ± 9.02	0.67 ± 0.02	55.42 ± 10.16
Media slice di tutti i pazienti	0.61 ± 0.02	41.61 ± 3.22	0.6 ± 0.02	41.24 ± 13.51	0.61 ± 0.01	42.37 ± 13.19	0.62 ± 0.02	42.52 ± 13.52

Tabella 7: Risultati (DSC e JAC) Test set 2

Tabella 8: Risultati (Accuratezza, Specificità, Sensibilità) ottenuti sul Test Set 2 senza Data Augmentation

Testing set: immagini ottenute da operatore manuale						
N° Pazienti	No Data Augmentation					
	No Post-processing			Post-processing		
	Accuratezza (%)	Specificità (%)	Sensibilità (%)	Accuratezza (%)	Specificità (%)	Sensibilità (%)
14	98,22 ± 1.35	98,2 ± 1,38	99,68 ± 0,7	97.85 ± 1.81	97.82 ± 1.85	99.84 ± 0.60
15	97,65 ± 2,07	97,66 ± 2,18	93,67 ± 15,23	97.12 ± 3.1	97.07 ± 3.25	97.68 ± 14.41
16	97,47 ± 1,4	97,54 ± 1,48	96,61 ± 14,5	96.8 ± 2.30	96.84 ± 2.4	94.06 ± 14.56
22	98,08 ± 0,57	98,11 ± 0,59	94,7 ± 9,4	97.96 ± 0.76	97.97 ± 0.78	95.65 ± 8.98
25	97.31 ± 9.53	97.36 ± 9.53	93.9 ± 14.77	97.14 ± 9.55	97.16 ± 9.56	95.33 ± 14.37
27	99.17 ± 0.28	99.22 ± 0.34	89.21 ± 28.0	99.19 ± 0.29	99.23 ± 0.34	89.2 ± 28.07
28	98.41 ± 0.55	98.43 ± 0.6	95.14 ± 14.44	98.42 ± 0.54	98.43 ± 0.59	95.41 ± 14.51
29	98.61 ± 0.47	98.92 ± 0.70	79.47 ± 39.77	98.55 ± 0.61	98.85 ± 0.84	79.82 ± 39.91
31	99.11 ± 0.31	99.17 ± 0.34	87.93 ± 19.66	99.11 ± 0.33	99.16 ± 0.37	89.12 ± 19.93
32	98.29 ± 0.57	98.41 ± 0.51	94.88 ± 11.04	98.33 ± 0.55	98.37 ± 0.55	96.95 ± 10.2
Media tutti i pazienti	98.35 ± 3.41	98.42 ± 3.42	92.18 ± 20.5	98.23 ± 3.54	98.27 ± 3.56	92.95 ± 20.47

Tabella 9: Risultati (Accuratezza, Specificità, Sensibilità) ottenuti sul Test set 2 con Data Augmentation

Testing set: immagini ottenute da operatore manuale						
N° Pazienti	Data Augumentation					
	No Post-processing			Post-processing		
	Accuratezza (%)	Specificità (%)	Sensibilità (%)	Accuratezza (%)	Specificità (%)	Sensibilità (%)
14	97.96± 1.29	97.93± 1.31	99.93± 0.30	98.32 ± 1.13	98.3 ± 1.15	99.88 ± 0.38
15	97.46 ± 2.28	97.45± 2.41	96.96± 14.43	97.81 ± 1.50	97.79 ± 1.60	97.6 ± 14.4
16	97.11± 1.78	97.17 ± 1.87	93.18± 15.89	97.41± 1.51	97.46 ± 1.59	93.98± 14.94
22	97.96 ± 0.69	97.98 ± 0.72	95.4 ± 9.08	98.17± 0.54	98.19± 0.56	95.61± 9.07
25	97.18 ± 9.55	97.18 ± 9.56	95.86± 14.03	97.35± 9.53	97.36± 9.53	96.2± 13.99
27	99.27 ± 0.28	99.32 ± 0.34	89.55± 28.07	99.21± 0.25	99.24± 0.33	90.42± 28.22
28	98.4 ± 0.58	98.41 ± 0.64	95.45± 14.52	98.42± 0.56	98.44± 0.62	95.46± 14.52
29	98.57± 0.59	98.88 ± 0.81	79.96± 39.98	98.64± 0.45	98.95± 0.68	79.93± 39.96
31	99.16 ± 0.37	99.19 ± 0.39	91.96± 16.64	99.1 ± 0.34	99.13± 0.37	92.64± 16.9
32	98.4 ± 0.54	98.47 ± 0.53	95.9± 10.63	97.44 ± 1.04	97.89± 0.74	71.76± 30.28
Media tutti i pazienti	98.31 ± 3.46	98.36 ± 3.48	93.11 ± 20.2	98.39 ± 3.39	98.42 ± 3.41	93.8 ± 20.16

Tabella 10: PPV ottenuto sul Test Set 2

Testing set: immagini ottenute da operatore manuale				
N° Pazienti	No Data Augumentation		Data Augumentation	
	No Post-processing	Post-processing	No Post-processing	Post-processing
	PPV (%)	PPV (%)	PPV (%)	PPV (%)
14	43.0 ± 10.72	50.02 ± 11.62	39.49 ± 8.35	53.95 ± 10.5
15	52.13 ± 14.67	48.43 ± 14.61	50.93%± 13.83	48.51 ± 14.06
16	36.5 ± 9.28	41.53 ± 11.0	34.64 ± 9.76	44.28 ± 10.61
22	44.67 ± 6.78	45.59 ± 7.34	43.56 ± 6.04	46.53 ± 7.03
25	43.21 ± 9.70	43.47 ± 9.97	42.91 ± 9.42	44.93 ± 9.62
27	43.62 ± 15.88	52.15 ± 16.0	47.14 ± 16.15	52.99 ± 16.13
28	37.33 ± 7.41	32.33 ± 7.58	37.32 ± 6.86	33.15% ± 7.55
29	32.35 ± 17.6	36.53 ± 17.21	79.96 ± 39.98	37.79 ± 17.89
31	34.73 ± 7.01	37.61 ± 6.82	37.32 ± 6.99	37.67 ± 6.78
32	55.28 ± 10.17	43.87 ± 10.44	56.7 ± 9.75	41.68 ± 19.27
Media tutti i pazienti	42.4 ± 13.3	43.87 ± 13.55	43.08 ± 13.36	45.12 ± 13.47

I pazienti 22 e 32 nei quali l'algoritmo semi-automatico ha rimosso correttamente il gesso, presentano performance pressoché uguali o lievemente migliori nel test set 2, in ogni caso si può ipotizzare che i risultati ottenuti con i due metodi siano praticamente uguali.

In generale, quindi, è evidente che le prestazioni della rete migliorano significativamente nelle slice in cui il gesso è stato rimosso rispetto a quelle in cui è ancora presente. Dal confronto tra la rete testata sull'intero set di test e sui singoli soggetti, emerge chiaramente che i soggetti nei quali il gesso non è stato rimosso mostrano prestazioni inferiori. Tuttavia, questi valori migliorano notevolmente quando il gesso viene rimosso manualmente.

L'algoritmo semi-automatico di rimozione del gesso si dimostra efficace in immagini standard o con poco rumore. Dai risultati dei test è emerso che, per i soggetti nei quali il gesso è stato rimosso dall'algoritmo, le prestazioni della rete sono pressoché equivalenti a quelle ottenute con la rimozione manuale del gesso. Questo indica che, quando l'algoritmo semi-automatico riesce a rimuovere correttamente il gesso, questo può sostituire efficacemente quello manuale.

La rete non raggiunge prestazioni eccessivamente elevate in termini di DSC e JAC eccetto in alcuni soggetti dove il DSC arriva anche a 0.7, in quanto i dati a disposizione per il training sono pochi e quindi la rete non è in grado di segmentare qualsiasi casistica ma solo i casi molto simili a quelli standard.

Tuttavia, mettendo a confronto le immagini presenti in Figura 46 e Figura 47 si nota che la frattura viene segmentata nel modo corretto in tutte le prove. Questa rete è in grado, quindi, di identificare le fratture di ulna, radio e scafoide.

In Figura 46 sono illustrate le maschere ottenute dalle slice in cui il gesso è stato rimosso dall'algoritmo semi-automatico, mentre in Figura 47 si possono apprezzare i risultati ottenuti sulle immagini modificate dall'operatore. Nel secondo le maschere ottenute sono nettamente migliori nella prima e terza riga, mentre le maschere della riga centrale sono uguali in quanto appartenenti al paziente 32 in cui la rimozione del gesso è avvenuta correttamente in entrambi i casi.

In Figura 48 e Figura 49 sono illustrate le rispettive segmentazioni. I soggetti rappresentati sono il 14, 32 e 28 rispettivamente nella prima, seconda e terza riga.

Maschera Manuale	Maschera Predetta			
	No DA – No Post	No DA – Post	DA – No Post	DA – Post
				
				
				

Figura 46: Risultati maschere ottenute con la rete neurale Test set 1

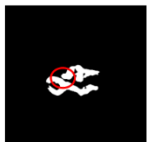
Maschera Manuale	Maschera Predetta			
	No DA – No Post	No DA – Post	DA – No Post	DA – Post
				
				
				

Figura 47: Risultati maschere ottenute con la rete neurale Test set 2

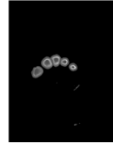
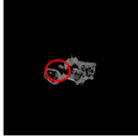
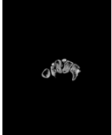
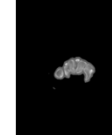
Segmentazione Manuale	Segmentazione ottenuta con le maschere predette			
	No DA – No Post	No DA – Post	DA – No Post	DA – Post
				
				
				

Figura 48- Segmentazioni ottenute con le maschere predette della rete neurale Test set 1

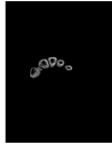
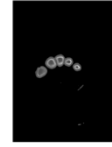
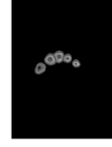
Segmentazione Manuale	Segmentazione ottenuta con le maschere predette			
	No DA – No Post	No DA – Post	DA – No Post	DA – Post
				
				
				

Figura 49- Segmentazioni ottenute con le maschere predette della rete neurale Test set 2

6 Conclusioni e sviluppi futuri

In base ai risultati ottenuti si può dedurre che il framework complessivo implementato, algoritmo di segmentazione semi-automatica e algoritmo di segmentazione neurale in cascata, funziona ma è limitato dal fatto che la qualità delle immagini è molto scarsa. Ciò è influenzato dall'elevato spessore delle slice, di media 2 mm, e di conseguenza anche il numero di slice per ogni TC risulta ridotto, in media 70.

È difficile individuare quale configurazione della rete U-Net sia la migliore, a causa della scarsa qualità, della ridotta variabilità e della limitata quantità di immagini nel campione (1501 per il training e 501 per la validazione), che rendono i risultati ottenuti con le diverse configurazioni non del tutto coerenti. Nello specifico, sono state testate diverse configurazioni della rete U-Net, in cui Data Augmentation e post-processing sono stati applicati in modo combinato, sia singolarmente che congiuntamente, così da valutare l'effetto di ciascuna tecnica e delle loro interazioni.

Durante la fase di training, le configurazioni migliori sono risultate quelle allenate senza Data Augmentation, mentre in fase di test le reti più bilanciate sono quelle che l'hanno utilizzata. Questo comportamento può essere attribuito a diversi fattori, tra cui il rischio di overfitting: con un campione ridotto, la rete può tendere a memorizzare i dettagli specifici dei dati di training, perdendo la capacità di generalizzare. L'introduzione della Data Augmentation in fase di addestramento aumenta la variabilità del dataset e può contribuire a rendere la rete più robusta nei confronti di dati non visti, simulando differenze tra training e test set e migliorando così le prestazioni complessive. In questo modo si riduce l'overfitting, ottenendo risultati più stabili e affidabili sui dati di test.

L'algoritmo di segmentazione semi-automatica al contrario permette di ottenere un modello 3D del polso in cui la frattura sia ben distinguibile, ne viene mostrato un esempio in Figura 50.

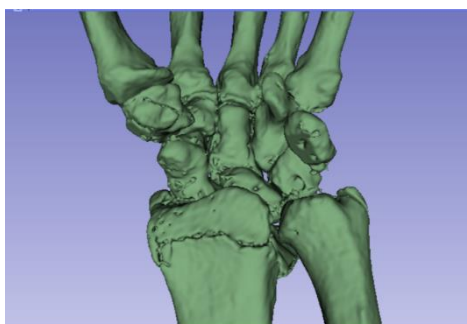


Figura 50: Modello 3D di frattura

Il processo complessivo, costituito quindi dalla rimozione del gesso tramite un algoritmo semi-automatico che utilizza tecniche tradizionali di CV e dalla successiva combinazione dell'implementazione di un modello di deep learning capace di creare delle maschere e segmentare l'osso, permette di ridurre i tempi di realizzazione delle maschere da parte dell'operatore poiché non è necessario effettuare le segmentazioni manuali complete ma solo modificarne alcune. Inoltre, la rimozione del gesso in maniera semi-automatica permette di creare le immagini da dare in input alla rete neurale senza che l'operatore debba eliminare il gesso manualmente con un software. Quindi riduce anche la dipendenza dalla mano dell'operatore e il lavoro che deve svolgere.

Tuttavia, sono presenti diverse limitazioni.

Per quanto riguarda l'algoritmo semi-automatico, prima di tutto rimane comunque ancora operatore-dipendente a causa di fattori come l'impostazione manuale delle soglie o la rifinitura delle maschere da dare in input alla rete. A causa della scarsa qualità delle immagini sia con l'algoritmo semi-automatico sia con la rete neurale non si ottengono segmentazioni di elevata qualità. Inoltre, tutte le soglie impostate sono state selezionate mediante metodo "*trail and error*" e quindi customizzate in base al campione preso in esame.

L'algoritmo di deep learning è limitato da un dataset molto ridotto e con poca variabilità. L'operatore che ha ritoccato le maschere del training e definito le maschere del testing non è un medico.

Le limitazioni attuali rappresentano opportunità significative per l'ulteriore sviluppo e perfezionamento dell'algoritmo in futuro.

Attualmente, durante la fase di pre-processing dell'algoritmo semi-automatico, il processo di *cropping* delle immagini è dipendente dall'operatore poiché la soglia per individuare la ROI contenente l'oggetto di interesse viene selezionata manualmente. Sebbene sia stata scelta una soglia che rappresenta un buon compromesso per tutte le immagini, per migliorare questo processo è necessario condurre uno studio su un campione adeguato di soggetti al fine di valutare l'intensità dei pixel dello sfondo. Uno degli obiettivi futuri è quello di sviluppare un metodo automatico in grado di analizzare queste intensità e riconoscere quali appartengono allo sfondo e non all'oggetto, così da poterle eliminare in modo efficiente. Inoltre, sarebbe utile ottimizzare l'algoritmo in modo da riuscire a riconoscere solo la regione anatomica di interesse escludendo tutte le altre presenti all'interno dell'immagine TC.

Questo approccio non solo migliorerebbe l'accuratezza del *cropping*, ma renderebbe anche il processo più robusto e ripetibile, riducendo la variabilità dovuta alla selezione manuale della soglia.

Anche il processo di rimozione di oggetti piccoli richiede l'impostazione manuale di una soglia dopo una serie di prove e un'analisi dei risultati ottenuti. Con un campione più ampio e variegato, sarebbe possibile condurre uno studio sulle dimensioni degli oggetti di interesse al fine di individuare una soglia generalizzata.

Infine, per la segmentazione dell'osso, il confronto tra due maschere generate da due algoritmi si basa su una soglia sperimentale, rendendo il processo dipendente dall'operatore. Testare l'algoritmo su un dataset appropriato consentirà di ottimizzare il processo di segmentazione finale, eliminando la necessità della soglia sperimentale e sviluppando un unico algoritmo in grado di produrre una maschera univoca per la segmentazione dell'osso.

Automatizzando queste fasi, si potrebbe risparmiare tempo e rendere il processo meno dipendente dall'operatore e più adattabile a diverse situazioni e tipologie di immagini.

Allenando le reti neurali su un dataset più ampio e variabile e con immagini di buona qualità è possibile ottenere delle segmentazioni precise. Potrebbe essere utile applicare diverse le tecniche di post-processing o considerare l'uso di diverse strategie di Data Augmentation e post-processing in combinazione.

L'implementazione di questi accorgimenti e la risoluzione delle criticità sopra elencate, permetterebbe di ottenere un framework in grado di riprodurre un modello 3D dalle immagini TC sul quale siano ben distinguibili le parti anatomiche e soprattutto in cui si possano identificare le fratture. In questo modo sarebbe possibile in cascata aggiungere un classificatore di frattura che lavori direttamente sui modelli 3D.

Uno sviluppo promettente è l'integrazione in un software di guida chirurgica intraoperatoria basato su realtà aumentata (AR). L'idea principale è sfruttare la potenza della realtà aumentata per creare un sistema che non solo aiuti i chirurghi durante gli interventi, ma che possa anche migliorare la pianificazione pre-operatoria e la gestione post-operatoria dei pazienti. Questo software permetterebbe di ridurre i tempi operatori grazie alla visualizzazione in tempo reale della struttura del polso e delle aree di interesse, consentendo una navigazione più intuitiva e precisa. Inoltre, considerando la complessità dell'articolazione del polso, il sistema può aiutare a minimizzare l'invasività dell'intervento, limitando i danni ai tessuti circostanti. Consentirebbe la previsione con maggiore precisione del risultato dell'intervento, facilitando così il monitoraggio post-operatorio e riducendo il numero di esami necessari nel follow-up. Questo comporterà una diminuzione dei costi per le strutture sanitarie e migliorerà l'esperienza del paziente. L'integrazione di un sistema di guida chirurgica intraoperatoria in AR ridurrebbe i tempi di attesa, poiché con interventi più rapidi e meno complicati aumenta

il numero di operazioni eseguibili e la possibilità di evitare interventi non necessari grazie alla previsione accurata del post-operatorio. Si ridurrebbe, così, anche il rischio operatorio.

Inoltre, l'AR offre significative opportunità per l'educazione e la formazione chirurgica, consentendo la simulazione di scenari realistici e l'interazione con modelli virtuali di ossa, migliorando l'apprendimento e creando un ambiente sicuro per lo sviluppo delle competenze professionali.

Un ulteriore avanzamento consiste nello sviluppo di un sistema basato su IA capace di determinare automaticamente la configurazione ottimale di un insieme di marker per guidare il medico durante l'intervento. Quindi, partendo dalla ricostruzione 3D del polso, l'IA potrebbe analizzare la frattura e suggerire i punti su cui intervenire e su quali posizionare le placche, ottimizzando così l'efficacia della guida intraoperatoria.

7 Lista di figure

<i>Figura 1: Anatomia articolazione del polso.....</i>	<i>8</i>
<i>Figura 2: Flowchart generale delle metodologie.....</i>	<i>21</i>
<i>Figura 3: Modello 3D del paziente 1. a) vista assiale; b) scansione completa</i>	<i>23</i>
<i>Figura 4: Viste scansione completa e assiale del soggetto 1 in 3D Slicer</i>	<i>23</i>
<i>Figura 5: Distribuzione soggetti in base al sesso</i>	<i>24</i>
<i>Figura 6: Distribuzione soggetti per età</i>	<i>24</i>
<i>Figura 7: Distribuzione fratture all'interno del dataset.....</i>	<i>25</i>
<i>Figura 8: Flowchart segmentazione manuale con algoritmo con tecniche di elaborazione delle immagini..</i>	<i>26</i>
<i>Figura 9: Workflow generale algoritmo di eliminazione gesso.....</i>	<i>27</i>
<i>Figura 10: Flowchart Data Preparation.....</i>	<i>28</i>
<i>Figura 11: Import file DICOM in 3D Slicer.....</i>	<i>28</i>
<i>Figura 12: Confronto immagine originale - import in Python.....</i>	<i>29</i>
<i>Figura 13: Flowchart Pre-Processing.....</i>	<i>29</i>
<i>Figura 14: Confronto immagine originale e croppata</i>	<i>31</i>
<i>Figura 15: Confronto immagine croppata e ridimensionata.....</i>	<i>31</i>
<i>Figura 16: Confronto tra immagine normalizzata e filtrata</i>	<i>32</i>
<i>Figura 17: Immagini risultati dagli step di Pre-processing.....</i>	<i>33</i>
<i>Figura 18: Flowchart processo Rimozione Gesso</i>	<i>34</i>
<i>Figura 19: Esempio applicazione filtro di Sobel.....</i>	<i>36</i>
<i>Figura 20: Esempio maschera ottenuta con l'applicazione del thresholding</i>	<i>36</i>
<i>Figura 21: Esempio immagine ottenuta con l'applicazione del flood fill ed eliminazione gesso.....</i>	<i>38</i>
<i>Figura 22: Esempio immagine ottenuta con applicazione di fill holes</i>	<i>38</i>

<i>Figura 23: Confronto tra le maschere ottenute dallo step 1 e lo step 2 del processo di rimozione gesso</i>	<i>40</i>
<i>Figura 24: Esempio immagine finale dopo rimozione gesso</i>	<i>41</i>
<i>Figura 25: Rimozione gesso - Algoritmo completo.....</i>	<i>41</i>
<i>Figura 26: Flowchart processo di segmentazione dell'osso.....</i>	<i>42</i>
<i>Figura 27: Immagini maschere pre-chiusura.....</i>	<i>43</i>
<i>Figura 28: Esempio maschere ottenute con applicazione flood fill</i>	<i>44</i>
<i>Figura 29: Esempio maschera finale</i>	<i>44</i>
<i>Figura 30: Confronto tra Maschera 1 e Maschera 2</i>	<i>47</i>
<i>Figura 31: Confronto segmentazione ottenuta dall'algoritmo completo e ridotto</i>	<i>49</i>
<i>Figura 32: Ricostruzione slice pre-processate con gesso rimosso, prima e dopo la ricostruzione</i>	<i>50</i>
<i>Figura 33: Ricostruzione volume soggetto 41.....</i>	<i>51</i>
<i>Figura 34: Flowchart Segmentazione con rete neurale</i>	<i>52</i>
<i>Figura 35: Confronto test set</i>	<i>55</i>
<i>Figura 36: Distribuzione fratture nel training set.....</i>	<i>56</i>
<i>Figura 37: Distribuzione fratture nel testing set.....</i>	<i>56</i>
<i>Figura 38: Soggetto 2.....</i>	<i>57</i>
<i>Figura 39: Soggetto 9.....</i>	<i>57</i>
<i>Figura 40: Soggetto 30.....</i>	<i>58</i>
<i>Figura 41: Schema rete per segmentazione U-Net.....</i>	<i>59</i>
<i>Figura 42: Addestramenti rete neurale</i>	<i>62</i>
<i>Figura 43: Prove test set</i>	<i>63</i>
<i>Figura 44: Esempio maschera manuale e maschera predetta</i>	<i>63</i>
<i>Figura 45: Confronto ricostruzione 3D del polso ottenuta con le segmentazioni generate dalla rete neurale (b) e quella ottenuta dalle immagini originali (a).....</i>	<i>66</i>

<i>Figura 46: Risultati maschere ottenute con la rete neurale Test set 1</i>	<i>82</i>
<i>Figura 47: Risultati maschere ottenute con la rete neurale Test set 2</i>	<i>82</i>
<i>Figura 48- Segmentazioni ottenute con le maschere predette della rete neurale Test set 1</i>	<i>83</i>
<i>Figura 49- Segmentazioni ottenute con le maschere predette della rete neurale Test set 2</i>	<i>83</i>
<i>Figura 50: Modello 3D di frattura</i>	<i>84</i>

8 Lista di tabelle

<i>Tabella 1: Risultati ottenuti sul training set</i>	<i>67</i>
<i>Tabella 2: Risultati ottenuti sul Validation set.....</i>	<i>68</i>
<i>Tabella 3: Risultati (DSC e JAC) ottenuti con il Test set 1</i>	<i>70</i>
<i>Tabella 4: Risultati (Accuratezza, Specificità, Sensibilità) ottenuti senza Data Augumentation</i>	<i>71</i>
<i>Tabella 5: Risultati (Accuratezza, Specificità, Sensibilità) ottenuti con Data Augumentation</i>	<i>72</i>
<i>Tabella 6: PPV ottenuto sul Test set 1</i>	<i>73</i>
<i>Tabella 7: Risultati (DSC e JAC) Test set 2</i>	<i>77</i>
<i>Tabella 8: Risultati (Accuratezza, Specificità, Sensibilità) ottenuti sul Test Set 2 senza Data Augumentation</i>	<i>78</i>
<i>Tabella 9: Risultati (Accuratezza, Specificità, Sensibilità) ottenuti sul Test set 2 con Data Augumentation ..</i>	<i>79</i>
<i>Tabella 10: PPV ottenuto sul Test Set 2.....</i>	<i>80</i>
<i>Tabella 11: Descrizione dataset.....</i>	<i>95</i>

9 Lista di abbreviazioni

AR	Realtà aumentata
CV	Computer Vision
DL	Deep learning
DSC	Dice similarity coefficient
IA	Intelligenza Artificiale
JAC	Indice di Jaccard
PPV	Valor predittivo positivo
ROI	Regione di Interesse
TC	Tomografia computerizzata
TR	Training set
TS	Test Set
VS	Validation Set

10 Appendice

10.1 Funzioni

1. **Crop_image(img, threshold):** la funzione prende in input l'immagine e la soglia di intensità al di sotto della quale i pixel vengono considerati come sfondo. In output si avrà l'immagine croppata e il vettore che contiene le coordinate spaziali dell'immagine croppata rispetto a quella originale. La funzione applica la soglia ad ogni pixel contenuto nell'immagine e trova le coordinate della più piccola ROI che contiene l'oggetto.
2. **Manual_resize((cropped_img, H, B):** la funzione prende in input l'immagine a cui è stata applicato il cropping, H e B che sono rispettivamente massima altezza e massima larghezza calcolata tra le dimensioni delle immagini croppate. In output si ottiene un'immagine con le nuove dimensioni. la funzione aggiunge un certo numero di pixel alle immagini sia in altezza che in larghezza per avere immagini delle stesse dimensioni.
3. **KeepLargestComponent(image):** la funzione prende in input l'immagine e rimuove piccoli componenti mediante il comando `remove_small_objects` (OpenCV: Morphology) con minima dimensione pari a 500 e connettività pari a 1. In output si avrà l'immagine ripulita.
4. **count_object_areas(binary_image):** la funzione prende in input l'immagine binaria e restituisce un dizionario che contiene tutte le aree presenti. La funzione etichetta gli oggetti nell'immagine binaria (prende in considerazione solo i pixel adiacenti in direzione verticale e orizzontale), calcola le loro proprietà connesse e restituisce un dizionario che associa ciascun oggetto alla sua area.
5. **filter_object_areas(object_areas, min_area_ratio=0.1):** la funzione prende in input il dizionario in cui a ciascun oggetto è associata la sua area e percentuale minima dell'area massima che un oggetto deve avere per essere mantenuto. Restituisce un dizionario in cui sono presenti solo gli oggetti che superano una certa percentuale di area rispetto all'area massima. Infatti, la funzione filtra gli oggetti in base alle loro aree, mantenendo solo quelli la cui area è maggiore o uguale a una percentuale specificata rispetto all'area massima tra tutti gli oggetti. Per impostazione predefinita, `min_area_ratio` è 0.1, di default verranno mantenuti solo gli oggetti la cui area è almeno il 10% dell'area massima.

-
6. **create_filtered_binary_image(labeled_image, filtered_areas):** prende in input un'immagine etichettata e un dizionario contenente gli oggetti filtrati con le relative aree e restituisce un'immagine binaria filtrata in base alle etichette degli oggetti conservati. La funzione utilizza le etichette degli oggetti conservati per creare un'immagine binaria filtrata in cui solo gli oggetti desiderati sono evidenziati.
7. **reconstruction_slice(resized_img, vett_coord_crop, diff_H, diff_B, volume_originale):** la funzione prende in input:
- a. **resized_img:** l'immagine ridimensionata che deve essere ripristinata alle sue dimensioni originali
 - b. **volume_originale:** il volume di immagini originali non processate dal quale è stata croppata l'immagine ridimensionata.
 - c. **vett_coord_crop:** il vettore contenente le coordinate dei vertici dell'immagine croppata rispetto a quella originale.
 - d. **diff_H:** La differenza tra il numero di righe dell'immagine ridimensionata e quella croppata.
 - e. **diff_B:** La differenza tra il numero di colonne dell'immagine ridimensionata e quella croppata.

In output si avrà l'immagine ricostruita alle sue dimensioni originali. La funzione svolge il compito di ripristinare un'immagine croppata alle sue dimensioni originali, utilizzando le informazioni sulle coordinate del cropping e le differenze tra le dimensioni originali e quelle ridimensionate. Quindi, vengono calcolate le dimensioni dell'immagine croppata sottraendo le differenze **diff_H** e **diff_B** dalle dimensioni dell'immagine ridimensionata e con queste viene ricostruita l'immagine croppata. Vengono calcolate le coordinate dei vertici. L'area croppata viene, poi, inserita nell'immagine ricostruita alle coordinate corrispondenti.

10.2 Descrizione del dataset

Tabella 11: Descrizione dataset

N Paziente	Sesso	Età	Classificazione Fratture	Tipo di scansione	Set di utilizzo	N slice TC	Note
1	F	70	frattura dell'epifisi del radio	AX	Training	89	
2	M	24	frattura dell'epifisi del radio	AX	Scartato	85	TC troppo distante da quella standard
3	M	27	frattura scafoide, radio sx	AX	Training	49	
7	F	53	frattura epifisi distale	AX	Training	91	
8	F	75	frattura radio distale	AX	Training	63	
9	M	17	frattura dell'epifisi del radio	AX	Scartato	82	
13	M	34	frattura radio distale	AX	Training	116	
14	F	58	frattura radio distale	AX	Testing	26	
15	M	64	frattura pluriframmentaria scomposta epifisi distale di radio e ulna	AX	Testing	47	
16	F	57	frattura radio distale	AX	Testing	76	
17	F	64	frattura dell'epifisi del radio	AX	Training	59	
18	F	74	frattura radio distale e diastasi dello spazio scafo-lunate	AX	Training	69	
19	F	76	FRATTURA POLSO	AX	Training	117	
20	F	69	frattura pluriframmentaria scomposta articolare	AX	Training	50	

			radio distale + frattura composta testa ulna				
21	F	53	FRATTURA POLSO	AX	Training	55	
22	M	26	frattura radio distale	AX	Training	72	
23	F	55	frattura di radio distale	AX	Training	47	
24	M	50	frattura di radio distale	AX	Testing	55	
25	M	58	frattura di radio distale	AX	Training	106	
26	M	53	frattura di radio distale	AX	Training	74	
27	M	52	frattura di radio distale	AX	Testing	181	
28	M	44	frattura di radio distale	AX	Testing	143	
29	F	23	frattura di radio distale	AX	Testing	55	
30	F	51	frattura radio distale e stolide ulnare	AX	Scartato		Dati non estraibili
31	M	61	frattura articolare pluriframmentaria scomposta dell'epifisi distale di radio	AX	Testing	88	
32	F	57	Frattura radio distale	AX	Testing	113	
41	F	71	frattura dell'epifisi del radio	AX Completo	Training	542	Scansione completa del soggetto 1
46	M	53	Frattura radio distale	AX ompleto	Validation	498	Scansione completa del soggetto 26

10.3 Codici

MATLAB

- *Rename_img*: il codice ha la funzione di organizzare e rinominare file di immagini e maschere appartenenti a pazienti specifici, copiandoli da una cartella di origine a una cartella di destinazione. L'obiettivo principale è uniformare i nomi dei file e aggiungere prefissi identificativi per ciascun paziente.

Nella prima parte del codice, viene gestita la ridenominazione delle immagini. Inizialmente, vengono definite le cartelle di origine e destinazione. La cartella di origine contiene le immagini denominate "slice_yy" (yy è il numero della slice), mentre la cartella di destinazione è dove le immagini saranno salvate con il nuovo formato "PZ_xx_slice_yy". L'identificativo del paziente (xx) viene inserito manualmente dall'utente e utilizzato per creare un prefisso specifico per i nomi dei file.

Il codice genera quindi un elenco di tutti i file presenti nella cartella di origine. Per ciascun file, viene verificato se il nome segue il formato specifico "slice_yy". Se il nome del file è conforme, viene aggiunto uno zero davanti al numero della fetta per uniformare la lunghezza del numero. Successivamente, i file vengono copiati dalla cartella di origine alla cartella di destinazione. Durante questo processo, viene estratto il numero della fetta dal nome del file originale e viene generato un nuovo nome che include il prefisso del paziente. Infine, il file viene rinominato nella cartella di destinazione.

Nella seconda parte del codice, si eseguono gli stessi passaggi per i file delle maschere dei pazienti.

- *PNG2DICOM*: il codice MATLAB ha l'obiettivo di convertire immagini PNG, che rappresentano segmentazioni ossee, in immagini DICOM, mantenendo e adattando i metadati delle immagini DICOM originali. Il processo è suddiviso in due parti principali: la conversione delle immagini PNG su cui è stato effettuato il post-processing e la conversione delle immagini PNG in cui non è avvenuto. Selezione della Cartella del Paziente:

La cartella relativa al paziente di interesse viene selezionata dall'utente. Successivamente, viene definita la struttura delle cartelle necessarie per accedere ai file DICOM originali e per salvare i nuovi file DICOM. Le cartelle sono organizzate in modo tale da separare chiaramente i file originali da quelli elaborati e non elaborati.

Il programma conta il numero di immagini DICOM presenti nella cartella originale per determinare il numero di iterazioni necessarie per il processo di conversione. Questo passaggio è fondamentale per assicurare che tutte le immagini vengano elaborate correttamente.

Viene definito il percorso che contiene le immagini PNG elaborate, rappresentanti la segmentazione ossea del paziente. Inoltre, viene creata una cartella di output dove verranno salvate le immagini DICOM risultanti dalla conversione.

Per ogni immagine PNG, il programma recupera i metadati corrispondenti dall'immagine DICOM originale, i quali vengono copiati nella nuova immagine DICOM.

L'immagine PNG viene scritta in formato DICOM, associandola ai metadati e salvata nella cartella di output.

Per le immagini non processate il procedimento è identico a quello descritto per le immagini processate.

PYTHON

- *Suddivisione_Creazione_Dataset*: il codice è progettato per preparare e organizzare i dati del dataset, garantendo che siano pronti per essere utilizzati nell'addestramento e nella valutazione di un modello di deep learning.

Inizialmente, vengono definiti i percorsi dei file per le immagini e le maschere, specificando le directory in cui sono conservate le immagini originali e le relative maschere. Successivamente, il codice crea le cartelle necessarie per organizzare i dati, che includono directory per il training set, il validation set e, al loro interno, directory specifiche per le immagini e le maschere.

Il passo successivo prevede l'estrazione delle liste dei file. Il codice estrae i nomi dei file delle immagini e delle maschere presenti nelle directory specificate. Le liste dei file vengono ordinate in base all'identificatore del paziente e al numero di slice, mantenendo così la coerenza tra le immagini e le relative maschere. Per ciascuna immagine, viene creato un dizionario contenente il percorso dell'immagine stessa e il percorso della relativa maschera, necessario per associare correttamente ogni immagine alla sua segmentazione.

Il dataset viene quindi suddiviso in un training set e un validation set, con l'obiettivo di utilizzare una parte dei dati per l'addestramento del modello e una parte per la valutazione delle prestazioni. La divisione del dataset viene eseguita utilizzando il metodo di divisione dei dati fornito dalla libreria scikit-learn, con una proporzione prestabilita per il validation set rispetto al totale del dataset.

Infine, le immagini e le maschere vengono copiate nelle rispettive directory create dal codice, garantendo che i dati siano organizzati correttamente

11 Bibliografia

- Benediktsson, J. A., Palmason, J. A., & Sveinsson, J. R. (2005). Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3), 480–491. <https://doi.org/10.1109/TGRS.2004.842478>
- Carelsen, B., Haverlag, R., Ubbink, D. T., Luitse, J. S. K., & Goslings, J. C. (2008). Does intraoperative fluoroscopic 3D imaging provide extra information for fracture surgery? *Archives of Orthopaedic and Trauma Surgery*, 128(12), 1419–1424. <https://doi.org/10.1007/s00402-008-0740-5>
- Chen, C., Liu, B., Zhou, K., He, W., Yan, F., Wang, Z., & Xiao, R. (2021). CSR-Net: Cross-Scale Residual Network for multi-objective scaphoid fracture segmentation. *Computers in Biology and Medicine*, 137. <https://doi.org/10.1016/j.compbiomed.2021.104776>
- Chen, L., Zhou, X., Wang, M., Qiu, J., Liu, S., & Mao, K. (n.d.). *ARU-Net: Research and Application for Wrist Reference Bone Segmentation*.
- Dalal, S., Singh, J. P., Tiwari, A. K., & Kumar, A. (2024). Identification of COVID-19 with CT scans using radiomics and DL-based features. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 13(1). <https://doi.org/10.1007/s13721-024-00448-3>
- Eelbode, T., Bertels, J., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., & Blaschko, M. B. (2020). Optimization for Medical Image Segmentation: Theory and Practice When Evaluating With Dice Score or Jaccard Index. *IEEE Transactions on Medical Imaging*, 39(11), 3679–3690. <https://doi.org/10.1109/TMI.2020.3002417>
- Gauthier, M., Beaulieu, J. Y., Nichols, L., & Hannouche, D. (2022). Ulna hook plate osteosynthesis for ulna head fracture associated with distal radius fracture. *Journal of Orthopaedics and Traumatology*, 23(1). <https://doi.org/10.1186/s10195-022-00658-3>
- Gou, X., Rao, Y., Feng, X., Yun, Z., & Yang, W. (2019). Automatic segmentation of ulna and radius in forearm radiographs. *Computational and Mathematical Methods in Medicine*, 2019. <https://doi.org/10.1155/2019/6490161>
-

-
- Han, X., Fan, J., Zheng, Y., Wu, Y., Alwalid, O., Ding, C., Jia, X., Li, H., Zhang, X., Zhang, K., Li, Y., Liu, J., Guo, T., Ren, H., & Shi, H. (2023). Value of radiomics in differentiating synchronous double primary lung adenocarcinomas from intrapulmonary metastasis. *Journal of Thoracic Disease*, 15(7), 3685–3698. <https://doi.org/10.21037/jtd-23-133>
- Hendrix, N., Scholten, E., Vernhout, B., Bruijnen, S., Maresch, B., de Jong, M., Diepstraten, S., Bollen, S., Schalekamp, S., de Rooij, M., Scholtens, A., Hendrix, W., Samson, T., Ong, L. L. S., Postma, E., van Ginneken, B., & Rutten, M. (2021). Development and validation of a convolutional neural network for automated detection of scaphoid fractures on conventional radiographs. *Radiology: Artificial Intelligence*, 3(4). <https://doi.org/10.1148/ryai.2021200260>
- Kornilov, A. S., & Safonov, I. V. (2018). An overview of watershed algorithm implementations in open source libraries. In *Journal of Imaging* (Vol. 4, Issue 10). MDPI. <https://doi.org/10.3390/jimaging4100123>
- Lee, S., Kim, K. G., Kim, Y. J., Jeon, J. S., Lee, G. P., Kim, K. C., & Jeon, S. H. (2024). Automatic Segmentation and Radiologic Measurement of Distal Radius Fractures Using Deep Learning. *CiOS Clinics in Orthopedic Surgery*, 16(1), 113–124. <https://doi.org/10.4055/cios23130>
- Lüdi, S., Kurz, C., Deforth, M., Ghafoor, H., Haefeli, M., & Honigmann, P. (2023). Radiological, Clinical, and Functional Outcomes of Combined Dorsal and Volar Locking Plate Osteosynthesis for Complex Distal Radius Fractures. *The Journal of Hand Surgery*, 48(4), 377–387. <https://doi.org/10.1016/J.JHSA.2021.12.002>
- Noguchi, S., Nishio, M., Yakami, M., Nakagomi, K., & Togashi, K. (2020). Bone segmentation on whole-body CT using convolutional neural network with novel data augmentation techniques. *Computers in Biology and Medicine*, 121. <https://doi.org/10.1016/j.compbiomed.2020.103767>
- Öztürk, C., Taşyürek, M., & Türkdamar, M. U. (2023). Transfer learning and fine-tuned transfer learning methods' effectiveness analyse in the CNN-based deep learning models. *Concurrency and Computation: Practice and Experience*, 35(4). <https://doi.org/10.1002/cpe.7542>
- Radke, K. L., Wollschläger, L. M., Nebelung, S., Abrar, D. B., Schleich, C., Boschheidgen, M., Frenken, M., Schock, J., Klee, D., Frahm, J., Antoch, G., Thelen, S., Wittsack, H. J., & Müller-lutz, A. (2021). Deep
-

learning-based post-processing of real-time mri to assess and quantify dynamic wrist movement in health and disease. *Diagnostics*, 11(6). <https://doi.org/10.3390/diagnostics11061077>

- Rong, C., Zhang, Q., Zhu, S., Zhang, G., Zeng, J., Han, Q., & Guo, Y. (2024). 3D printed guide-assisted percutaneous screw fixation for minimally displaced scaphoid waist fractures with delayed diagnosis or presentation. *BMC Musculoskeletal Disorders*, 25(1). <https://doi.org/10.1186/s12891-024-07243-1>
- Ryan, P. J., Duckworth, A. D., Mceachan, J. E., & Jenkins, P. J. (n.d.). *The incidence of surgical intervention following a suspected scaphoid fracture*. <https://doi.org/10.1302/2633-1462>
- Shen, Z., wang, H., Shao, Y., Duan, Y., Gu, H., Chen, H., Feng, A., Huang, Y., & Xu, Z. (2023). Optimization of isocenter position for multiple brain metastases single-isocenter stereotactic radiosurgery to minimize dosimetric variations due to rotational uncertainty. *Physica Medica*, 111. <https://doi.org/10.1016/j.ejmp.2023.102614>
- Sigmund, O. (2009). Manufacturing tolerant topology optimization. *Acta Mechanica Sinica/Lixue Xuebao*, 25(2), 227–239. <https://doi.org/10.1007/s10409-009-0240-z>
- Srinivasan, S., Durairaju, K., Deebe, K., Mathivanan, S. K., Karthikeyan, P., & Shah, M. A. (2024). Multimodal Biomedical Image Segmentation using Multi-Dimensional U-Convolutional Neural Network. *BMC Medical Imaging*, 24(1). <https://doi.org/10.1186/s12880-024-01197-5>
- Staffas, T., Elshaari, A., & Zwiller, V. (2024). Frequency modulated continuous wave and time of flight LIDAR with single photons: a comparison. *Optics Express*, 32(5), 7332. <https://doi.org/10.1364/oe.508004>
- Su, L., Fu, X., Zhang, X., Cheng, X., Ma, Y., Gan, Y., & Hu, Q. (2018). Delineation of Carpal Bones from Hand X-Ray Images Through Prior Model, and Integration of Region-Based and Boundary-Based Segmentations. *IEEE Access*, 6, 19993–20008. <https://doi.org/10.1109/ACCESS.2018.2815031>
- Suganthi, N., Kotagiri, S., Thirupurasundari, D. R., & Vimala, S. (2024). Adam golden search optimization enabled DCNN for classification of breast cancer using histopathological image. *Biomedical Signal Processing and Control*, 94. <https://doi.org/10.1016/j.bspc.2024.106239>
-

Vardhana, M., Arunkumar, N., Lasrado, S., Abdulhay, E., & Ramirez-Gonzalez, G. (2018). Convolutional neural network for bio-medical image segmentation with hardware acceleration. *Cognitive Systems Research*, 50, 10–14. <https://doi.org/10.1016/j.cogsys.2018.03.005>