



UNIVERSITÀ DEGLI STUDI DI CASSINO E DEL LAZIO MERIDIONALE

Corso di Laurea Magistrale in Ingegneria Informatica

Segmentazione del seno in immagini MRI

Versione 0.1

Studente: Achille Cannavale

Matricola: 59721

Relatore: Alessandro Bria

6 luglio 2025

Indice

1	Introduzione	3
1.1	Contesto	3
1.2	Attività di Tirocinio e obiettivi	4
1.3	Strumenti Utilizzati	4
1.3.1	Connessione in SSH e GPU	5
2	Background e Stato dell'Arte	6
2.1	L'Evoluzione delle Tecniche di Segmentazione	6
2.2	Strumenti e Dataset Moderni	7
2.3	Le Sfide Aperte	7
3	Dataset: Duke Breast Cancer MRI	9
3.1	Composizione del Dataset	9
4	Esperimenti	11
4.1	Il file di configurazione YAML	12
4.1.1	Struttura generale	12
4.1.2	Configurazione del training	12
4.1.3	Trasformazioni dei dati	13
4.1.4	Definizione del modello	13
4.1.5	Ottimizzazione e loss	13
4.1.6	Post-processing e valutazione	14

Capitolo 1

Introduzione

Contents

1.1	Contesto	3
1.2	Attività di Tirocinio e obiettivi	4
1.3	Strumenti Utilizzati	4
1.3.1	Connessione in SSH e GPU	5

1.1 Contesto

La segmentazione di strutture anatomiche in immagini mediche 3D rappresenta un task importante nell'ambito clinico e diagnostico. In particolare la segmentazione del seno da immagini MRI (Magnetic Resonance Imaging) 3D è un task complesso, ma essenziale per applicazioni come la pianificazione di interventi chirurgici o l'analisi di anomalie tissutali.

In passato, la segmentazione del seno in immagini MRI 3D veniva effettuata principalmente con **metodi classici**, spesso semi-automatici o manuali, basati su:

- **Approcci a soglia** (thresholding) e **region-growing**, che sfruttavano differenze di intensità tra tessuti ma richiedevano regolazioni manuali e fallivano in presenza di rumore o basso contrasto.
- **Deformable models** (es.: Active Contours) e **algoritmi a clustering** (es.: k-means), sensibili all'inizializzazione e poco robusti alla variabilità anatomica.
- Metodi **atlas-based**, che allineavano immagini a template pre-annotati, limitati però dalla diversità inter-paziente.

Con l'avvento del **deep learning**, in particolar modo delle reti convoluzionali e di architetture come **U-Net 3D**, la segmentazione del seno ha raggiunto livelli di accuratezza più alti rispetto al passato.

1.2 Attività di Tirocinio e obiettivi

L'attività si è inserita nel contesto di un progetto di ricerca mirato allo sviluppo e alla valutazione di modelli di segmentazione automatica per immagini di risonanza magnetica della mammella, con particolare attenzione.

L'obiettivo principale del tirocinio è stato quello di sperimentare e ottimizzare pipeline basate su **deep learning** per la segmentazione semantica tridimensionale, utilizzando framework open-source moderni come **MONAI** (Medical Open Network for AI) e **PyTorch**.

Le principali attività svolte durante il tirocinio sono state:

- **Pre-processing** dei dati in formato DICOM, con conversione in formato leggibile da MONAI e normalizzazione dell'intensità del segnale.
- Composizione e validazione di un dataset bilanciato, con partizionamento in training, validation e test set.
- Studio e implementazione di modelli di segmentazione basati su architetture **state-of-the-art**, tra cui **UNet** e **AttentionUNet**.
- Configurazione degli esperimenti, tuning degli iperparametri e addestramento dei modelli in ambiente GPU.
- Valutazione delle prestazioni mediante metriche standard come Dice coefficient

Durante il tirocinio sono state inoltre affrontate e risolte diverse problematiche tecniche legate alla gestione dei metadati DICOM, alla compatibilità tra i formati di input, e alla gestione efficiente della memoria in fase di training. L'intero lavoro è stato documentato e riproducibile mediante script Python e configurazioni YAML modulari.

1.3 Strumenti Utilizzati

Durante il tirocinio sono stati impiegati diversi strumenti software e librerie fondamentali per la gestione, il pre-processing e l'elaborazione di immagini medicali, nonché per lo sviluppo e il training di modelli di deep learning.

- Il linguaggio principale di lavoro è stato **Python**, scelto per la sua flessibilità e per l'ampio ecosistema di librerie scientifiche.
- Per la manipolazione delle immagini DICOM è stata utilizzata la libreria **MONAI** (Medical Open Network for AI), un framework open-source basato su PyTorch e progettato specificamente per applicazioni di imaging medicale. MONAI ha permesso di gestire agevolmente il caricamento dei dati, le trasformazioni e la normalizzazione delle immagini, grazie a un sistema modulare di trasformazioni componibili.
- Il framework di deep learning impiegato è stato **PyTorch**, scelto per la sua semplicità d'uso, il supporto attivo della community e le sue prestazioni elevate, specialmente in combinazione con l'accelerazione GPU fornita da CUDA.

- Lo sviluppo del codice è stato realizzato su una macchina Linux, con accesso remoto e con l'ausilio dell'editor **Visual Studio Code**.

1.3.1 Connessione in SSH e GPU

Durante il tirocinio, ho avuto accesso alla potenza di calcolo delle GPU universitarie, dedicate a progetti di ricerca in deep learning. Per sfruttare queste risorse, ho utilizzato il protocollo SSH (Secure Shell) per connettermi in remoto ai server e gestire l'esecuzione dei miei esperimenti.

Il sistema era dotato di 4 GPU NVIDIA (modello Tesla V100 o A100, a seconda della disponibilità), condivise tra diversi utenti del dipartimento. Per evitare conflitti nell'allocazione delle risorse, era necessario prenotare in anticipo le GPU tramite un sistema di schedulazione interno, basato su code di priorità.

Capitolo 2

Background e Stato dell'Arte

Contents

2.1	L'Evoluzione delle Tecniche di Segmentazione	6
2.2	Strumenti e Dataset Moderni	7
2.3	Le Sfide Aperte	7

La segmentazione semantica di immagini mediche rappresenta una delle sfide più importanti nel campo della diagnostica assistita da computer. Questo compito, che consiste nel delineare con precisione strutture anatomiche o aree patologiche all'interno di immagini biomediche, ha un impatto diretto su applicazioni cliniche cruciali come la pianificazione chirurgica, il monitoraggio terapeutico e la radiomica. Nel contesto specifico della mammella, la segmentazione da immagini MRI 3D presenta peculiarità che la rendono particolarmente complessa, tra cui l'elevata variabilità anatomica tra pazienti, la presenza di artefatti tipici delle risonanze magnetiche e la necessità di bilanciare accuratezza e tempi di elaborazione quando si lavora con volumi tridimensionali ad alta risoluzione.

2.1 L'Evoluzione delle Tecniche di Segmentazione

Prima dell'avvento del deep learning, la segmentazione di immagini mediche si basava principalmente su approcci tradizionali che, pur rappresentando soluzioni pionieristiche per l'epoca, presentavano limiti significativi. Tecniche come il thresholding e il region-growing, ad esempio, erano ampiamente utilizzate per la loro semplicità concettuale, ma risultavano estremamente sensibili alla qualità dell'immagine, fallendo spesso in presenza di rumore o basso contrasto tra i tessuti. Allo stesso modo, i deformable models, che cercavano di adattare contorni attivi alle strutture anatomiche, richiedevano un'inizializzazione manuale e faticavano a gestire la complessa morfologia della ghiandola mammaria.

Con l'introduzione delle reti neurali convoluzionali (CNN), il panorama della segmentazione medica è cambiato radicalmente. L'architettura U-Net, proposta nel 2015 da Ronneberger et

al., ha rappresentato una svolta grazie alla sua struttura encoder-decoder e alle connessioni skip, che permettono di combinare informazioni a diversi livelli di risoluzione, preservando i dettagli spaziali fondamentali per una segmentazione precisa. Successivamente, la comunità scientifica ha sviluppato varianti sempre più avanzate, come la V-Net, ottimizzata per dati volumetrici, e il framework nnU-Net, in grado di adattarsi automaticamente alle caratteristiche di diversi dataset medici.

Negli ultimi anni, l'attenzione si è spostata verso i Transformers, modelli nati nell'ambito del Natural Language Processing e poi adattati con successo all'analisi di immagini. Architetture come TransUNet e Swin UNETR combinano la capacità delle CNN di estrarre features locali con il potere dei Transformers di modellare relazioni globali, offrendo prestazioni superiori in molti task di segmentazione. Tuttavia, questi modelli richiedono risorse computazionali elevate e grandi quantità di dati annotati, il che ne limita ancora l'applicabilità in alcuni contesti clinici.

2.2 Strumenti e Dataset Moderni

Oggi, lo sviluppo di pipeline per la segmentazione medica si avvale di strumenti sempre più sofisticati. Tra questi, il framework MONAI (Medical Open Network for AI) si è affermato come uno standard de facto, grazie alla sua vasta raccolta di trasformazioni specifiche per immagini mediche, modelli predefiniti e metriche di valutazione. Basato su PyTorch, MONAI semplifica notevolmente la gestione di dati complessi come quelli DICOM e supporta l'implementazione di workflow riproducibili, essenziali per la ricerca in ambito medico.

Per quanto riguarda i dataset, il Duke-Breast-Cancer-MRI, disponibile su The Cancer Imaging Archive (TCIA), rappresenta una risorsa preziosa per lo studio della segmentazione mammaria. Questo dataset include volumi MRI multiparametrici annotati manualmente da radiologi esperti, catturando tutta la complessità anatomica e le sfide tipiche delle immagini reali, come la presenza di lesioni e la variabilità nella densità del tessuto.

2.3 Le Sfide Aperte

Nonostante i progressi compiuti, la segmentazione del seno in MRI 3D presenta ancora diverse criticità. Una delle principali è la scarsità di dataset pubblici di grandi dimensioni, soprattutto se confrontata con la disponibilità di dati per altri organi come il cervello o il fegato. Inoltre, i modelli esistenti faticano spesso a generalizzare su dati provenienti da diversi centri medici, a causa delle variazioni tra scanner e protocolli di acquisizione.

Un altro aspetto critico è l'integrazione di questi strumenti nei workflow clinici quotidiani. Molti modelli, pur offrendo prestazioni elevate in contesti sperimentali, non sono ancora ottimizzati per l'uso in tempo reale o per interagire con i sistemi informativi ospedalieri. Infine, la mancanza di interpretabilità delle predizioni rimane un ostacolo significativo per l'adozione clinica, poiché i medici necessitano di comprendere le basi delle decisioni algoritmiche.

Questo contesto evidenzia l'importanza di sviluppare soluzioni innovative che affrontino non solo gli aspetti tecnici della segmentazione, ma anche le esigenze pratiche degli operatori

sanitari. La pipeline proposta in questo lavoro si inserisce proprio in questo spazio, cercando di colmare alcune delle lacune esistenti attraverso un approccio bilanciato tra accuratezza, efficienza e adattabilità.

Capitolo 3

Dataset: Duke Breast Cancer MRI

Contents

3.1	Composizione del Dataset	9
-----	--------------------------	---

3.1 Composizione del Dataset

Il dataset utilizzato durante il tirocinio proviene dal database **Duke Breast Cancer MRI**, un archivio pubblico contenente immagini DICOM di risonanze magnetiche della mammella. Ogni caso clinico è organizzato in una struttura gerarchica che riflette l'identificativo univoco del paziente e delle relative acquisizioni.

Il dataset è composto da sequenze **DICOM** tridimensionali, ciascuna rappresentante un'acquisizione volumetrica del seno. In tutti i casi sono disponibili anche annotazioni o maschere segmentate manualmente da esperti, utilizzate come ground truth per il training dei modelli di segmentazione.

Durante la fase di preparazione, le immagini sono state convertite in formato compatibile con MONAI tramite una pipeline di preprocessing, che ha incluso:

- il **caricamento** delle immagini in memoria;
- la standardizzazione dell'intensità e il ridimensionamento spaziale;
- l'organizzazione dei dati in **dizionari** contenenti sia l'immagine sia i relativi metadati;
- la **salvataggio** del dataset preprocessato

Per ogni volume è stata mantenuta l'associazione tra immagine e metadati (es. spacing, orientamento, posizione del paziente), necessari per garantire una corretta elaborazione e interpretazione dei dati durante le fasi di training, validazione e inferenza.

L'intero dataset è stato infine suddiviso in tre sottoinsiemi: **training**, **validation** e **test**, rispettando la proporzione e la varietà dei casi clinici per assicurare una valutazione affidabile delle performance del modello.

Capitolo 4

Esperimenti

Contents

4.1	Il file di configurazione YAML	12
4.1.1	Struttura generale	12
4.1.2	Configurazione del training	12
4.1.3	Trasformazioni dei dati	13
4.1.4	Definizione del modello	13
4.1.5	Ottimizzazione e loss	13
4.1.6	Post-processing e valutazione	14

Per effettuare gli esperimenti ho avuto la possibilità di utilizzare un framework sperimentale sviluppato da un dottorando del laboratorio. Il cuore del framework risiedeva nella sua capacità di astrarre la complessità tipica degli esperimenti di segmentazione 3D attraverso una struttura di configurazione gerarchica e intuitiva. Ogni aspetto cruciale del processo, dal preprocessing dei dati alla definizione dell'architettura, dagli iperparametri di training alle metriche di valutazione, trovava una precisa collocazione nel file YAML.

Uno dei maggiori vantaggi di questo approccio emergeva nella fase di validazione incrociata delle ipotesi. La possibilità di confrontare direttamente diverse varianti architetturelle, mantenendo invariati tutti gli altri parametri, ha permesso di isolare con precisione l'impatto di ciascuna scelta progettuale.

Sebbene il framework fornisse una struttura organizzata per gli esperimenti, non era dotato di sistemi intelligenti in grado di verificare automaticamente la coerenza logica delle configurazioni. Spettava quindi a me, durante la definizione dei parametri nel file YAML, assicurarmi che le scelte fossero compatibili tra loro e adatte al contesto. Ad esempio, se impostavo un'architettura progettata per input volumetrici, dovevo verificare manualmente che anche le trasformazioni di preprocessing—come il random cropping o il resizing—fossero configurate per operare su tre dimensioni, evitando incongruenze che avrebbero compromesso l'addestramento.

Allo stesso modo, quando sperimentavo ottimizzatori diversi, dovevo prestare attenzione alla scelta del learning rate e dello scheduler, poiché valori troppo aggressivi potevano destabilizzare il training, mentre quelli troppo conservativi rallentavano inutilmente la convergenza. Questo processo richiedeva una costante analisi degli errori: se un esperimento falliva o produceva risultati insoliti, dovevo riesaminare la configurazione YAML per identificare possibili discrepanze, come dimensioni di crop incompatibili con la risoluzione del volume o parametri di augmentazione eccessivamente distorti. La mancanza di validazione automatica ha reso il lavoro più impegnativo, ma mi ha costretto a sviluppare una profonda comprensione delle interdipendenze tra i vari componenti del sistema.

4.1 Il file di configurazione YAML

Il file di configurazione YAML è organizzato in sezioni logiche che definivano ogni aspetto del processo di addestramento e valutazione. Di seguito, analizziamo le componenti principali del file, suddividendolo in sottoinsiemi funzionali.

4.1.1 Struttura generale

Il file YAML è organizzato in 6 sezioni principali:

```
# Esempio di struttura generale
TRAINING:
  # parametri di addestramento
TRAIN_TRANSFORM:
  # trasformazioni per il training
TEST_TRANSFORM:
  # trasformazioni per il test
MODEL:
  # architettura del modello
LOSS:
  # funzione di loss
OPTIMIZER:
  # ottimizzatore
```

4.1.2 Configurazione del training

La sezione TRAINING definiva i parametri fondamentali:

```
TRAINING:
  workspace: /path/to/experiment_results
  dataset_root: /path/to/dataset
  valid_size: 0.1 # 10% validation set
  test_size: 0.1 # 10% test set
  train_batch_size: 4
  test_batch_size: 1
  num_workers: 8 # thread per data loading
```

```

device: cuda:0      # GPU da utilizzare
epochs: 50          # numero di epoche
roi_size: [256, 256, 32] # dimensione regioni di interesse

```

4.1.3 Trasformazioni dei dati

Le sezioni TRAIN_TRANSFORM e TEST_TRANSFORM specificavano il preprocessing:

TRAIN_TRANSFORM:

```

- class: monai.transforms.LoadImaged
  params:
    keys: ['post_2', 'breast']
- class: monai.transforms.RandSpatialCropSamplesd
  params:
    keys: ['img', 'seg']
    num_samples: 4
    roi_size: [512, 512, 8]

```

4.1.4 Definizione del modello

La sezione MODEL configurava l'architettura della rete:

MODEL:

```

class: package.networks.UNet3D
params:
  channels: [16, 32, 64] # canali per ogni livello
  in_channels: 1         # canali input
  out_channels: 2        # canali output
  norm: INSTANCE         # normalizzazione
  spatial_dims: 3        # dimensione spaziale
  strides: [2, 2, 2]     # stride dei livelli

```

4.1.5 Ottimizzazione e loss

Le sezioni LOSS e OPTIMIZER controllavano l'addestramento:

LOSS:

```

class: package.losses.CombinedLoss
params:
  weight_dice: 0.7
  weight_ce: 0.3

```

OPTIMIZER:

```

class: torch.optim.AdamW
params:
  lr: 1e-3                # learning rate
  weight_decay: 1e-3      # regolarizzazione

```

4.1.6 Post-processing e valutazione

Le sezioni finali gestivano la valutazione:

```
POST_PRED:
- class: monai.transforms.AsDiscrete
  params:
    argmax: true
    to_onehot: 2

METRIC:
class: monai.metrics.DiceMetric
params:
  include_background: false
```

