UNIVERSITÀ DEGLI STUDI DI CASSINO E DEL LAZIO MERIDIONALE

Corso di Laurea Magistrale in Ingegneria Informatica

Segmentazione del seno in immagini MRI

Studente: Achille Cannavale
Matricola: 59721
Relatore: Alessandro Bria

Indice

Capitolo 1

Introduzione

Contents

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 chirugici o l'analisi di anomalie tessutali.

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** [2], 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 in MRI del seno, con l'obiettivo di migliorare la diagnosi precoce e la pianificazione terapeutica.

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) [1] 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 delle immagini.
- 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 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 Score

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.

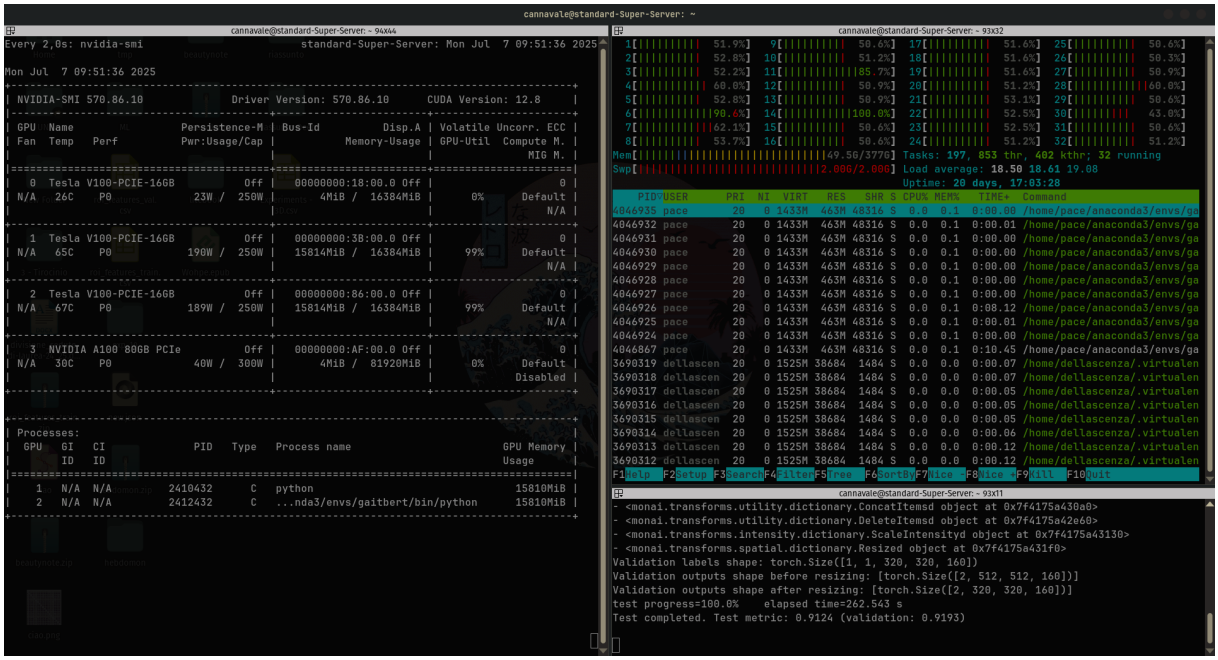
Il sistema era dotato di 4 GPU NVIDIA (modello Tesla V100 o A100, a seconda della disponibilità), condivise tra diversi utenti del dipartimento. Per evitare conflitti nell'allocazione delle risorse, è stato necessario prenotare in anticipo le GPU tramite un sistema di schedulazione interno, basato su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

Nel dettaglio, il nodo su cui ho lavorato disponeva delle seguenti GPU:

- **3x NVIDIA Tesla V100-PCIE-16GB**, GPU ad alte prestazioni con 16 GB di memoria, molto diffuse in ambito scientifico per il training di deep neural networks.
- **1x NVIDIA A100 80GB PCIe**, una delle GPU più potenti attualmente disponibili per il calcolo scientifico, con ben 80 GB di memoria dedicata.

Le informazioni mostrate da `nvidia-smi` includevano anche temperatura, consumo energetico, utilizzo della memoria e livello di attività di ciascuna GPU.

Questa infrastruttura ha permesso di testare i miei modelli su dataset di grandi dimensioni con tempi di addestramento molto più rapidi rispetto all'uso di una macchina locale.



Capitolo 2

Background e Stato dell'Arte

Contents

La segmentazione semantica di immagini mediche rappresenta una delle sfide più importanti nel campo della diagnostica. 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 e il monitoraggio terapeutico. 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**

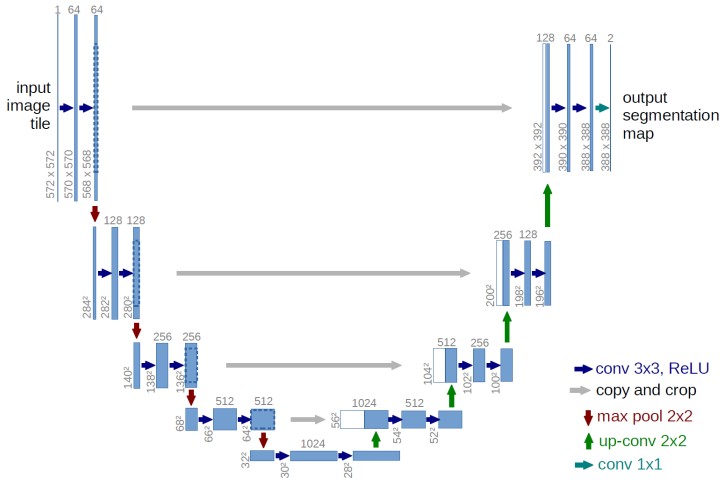
al., ha rappresentato una svolta grazie alla sua struttura **encoder-decoder** e alle **skip connections**, che permettono di combinare informazioni a diversi livelli di risoluzione, preservando i dettagli spaziali fondamentali per una segmentazione precisa. Successivamente, la comunità scientifica ha 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 limenta ancora l'applicabilità in alcuni contesti clinici.

2.2 Architettura U-Net per la Segmentazione di Immagini Mediche

Tra le architetture più utilizzate nel campo della segmentazione semantica applicata all'imaging biomedico, la **U-Net** occupa senza dubbio un ruolo di primo piano. Introdotta da Ronneberger et al. nel 2015, questa rete è stata progettata appositamente per segmentare immagini mediche anche in condizioni di disponibilità limitata di dati annotati, un aspetto ricorrente nei contesti clinici reali. Il successo della U-Net è dovuto non solo alla sua efficacia, ma anche alla sua semplicità architetturale, che la rende estremamente versatile e facilmente adattabile a diverse tipologie di dati, tra cui immagini 2D, 3D o multi-canale.

L'architettura della U-Net si sviluppa secondo una struttura simmetrica a forma di U, composta da due fasi principali: un percorso di contrazione, o encoder, e un percorso di espansione, o decoder. La fase di contrazione ha il compito di estrarre rappresentazioni sempre più astratte e semantiche dell'immagine in input, riducendone progressivamente la risoluzione attraverso convoluzioni e operazioni di pooling. Al contrario, la fase di espansione mira a ricostruire l'informazione spaziale originaria, riportando l'output alla dimensione dell'immagine di partenza mediante operazioni di upsampling e convoluzioni.



Una caratteristica distintiva della U-Net rispetto ad altre reti di segmentazione è la presenza delle cosiddette *skip connections*. Questi collegamenti diretti tra i livelli corrispondenti dell'encoder e del decoder permettono di trasportare le informazioni a bassa astrazione, spesso perse durante il downsampling, direttamente nei livelli di ricostruzione. In questo modo, la rete riesce a combinare efficacemente il contesto globale dell'immagine con i dettagli locali, migliorando significativamente la precisione dei bordi e la definizione delle strutture segmentate.

Dal punto di vista pratico, la U-Net prende in input un'immagine e produce come output una maschera segmentata, in cui ogni pixel (o voxel nel caso 3D) è classificato in una determinata categoria. Questo rende la rete particolarmente adatta per compiti dove è richiesta una segmentazione voxel-wise, come nel caso dell'identificazione di tessuti, lesioni o strutture anatomiche specifiche.

Nel progetto descritto in questa tesi, la U-Net è stata adottata come architettura di base per la segmentazione automatica del seno e del tessuto fibroghiandolare in immagini MRI. Questo approccio è ispirato anche al lavoro recente di Lew et al. (2024), in cui due reti U-Net sono state combinate in cascata per segmentare prima il seno intero e successivamente il tessuto interno, comprendente FGT e vasi sanguigni. In tal senso, la U-Net si è dimostrata una scelta solida, grazie alla sua capacità di generalizzare anche su strutture anatomiche complesse, pur mantenendo una buona efficienza computazionale.

Nonostante la sua efficacia, la U-Net presenta alcune limitazioni. In presenza di rumore, artefatti o grandi squilibri tra le classi, la rete può mostrare difficoltà, ad esempio nel riconoscere strutture piccole o nel distinguere tessuti contigui con caratteristiche simili. Per superare tali difficoltà, sono state sviluppate diverse estensioni e varianti dell'architettura originale, come l'U-Net++ con connessioni più profonde, le U-Net con meccanismi di attention, o le versioni 3D per dati volumetrici. Tuttavia, anche nella sua forma standard, la U-Net continua a rappresentare un punto di partenza solido e affidabile per molte applicazioni di segmentazione in ambito medico.

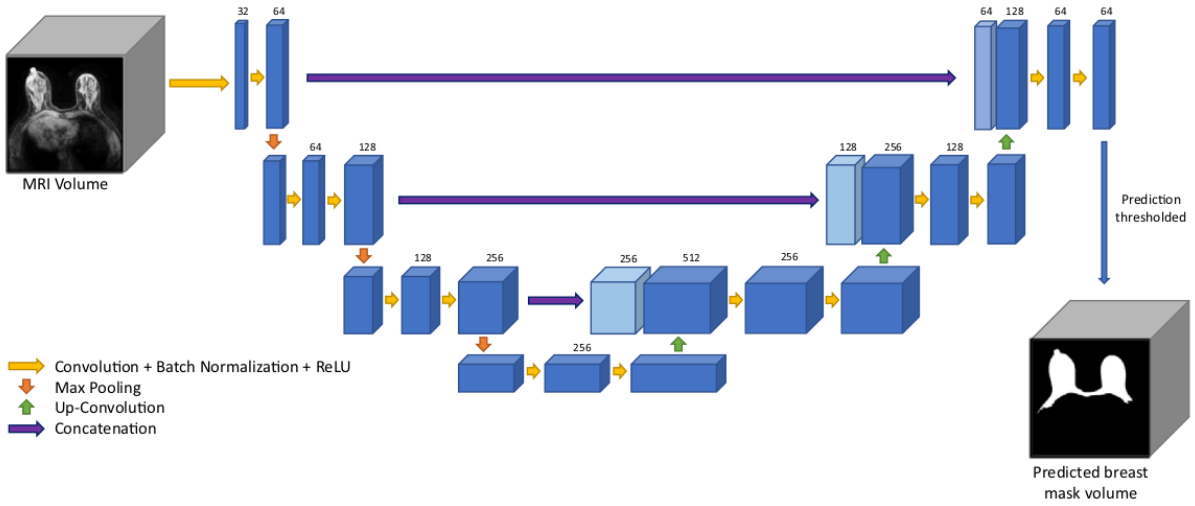
2.3 Strumenti e Dataset Moderni

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.

2.4 Stato dell'Arte nella Segmentazione del Seno in MRI 3D

Negli ultimi anni, la segmentazione automatica delle immagini mediche ha ricevuto un crescente interesse grazie all'utilizzo di reti neurali convoluzionali (CNN), in particolare per compiti complessi come l'analisi della densità mammaria. Uno studio di riferimento in questo ambito è quello di Lew et al. (2024) [4], che ha proposto un modello basato su U-Net per la segmentazione automatica del seno, del tessuto fibroghiandolare (FGT) e dei vasi sanguigni su risonanza magnetica (MRI) pre-contrasto.



Il lavoro si distingue per l'approccio rigoroso e riproducibile all'annotazione dei dati: 100 studi MRI, selezionati dal dataset pubblico *Duke Breast Cancer MRI*, sono stati annotati manualmente seguendo criteri ben definiti e validati da radiologi specializzati. Le annotazioni sono tridimensionali e comprendono il tessuto mammario, il FGT e i vasi, un elemento spesso trascurato in studi precedenti.

Dal punto di vista architetturale, il sistema utilizza due reti U-Net in cascata:

- **Breast U-Net**, che segmenta il tessuto mammario nell'intero volume MRI.
- **FGT-Vessel U-Net**, che utilizza sia il volume MRI che la segmentazione del seno come input per individuare il FGT e i vasi sanguigni.

Le performance sono state valutate tramite il coefficiente di similarità di Dice (DSC), ottenendo i seguenti risultati medi sul test set:

- **Breast segmentation: DSC = 0.92 (3D), 0.95 (2D)**
- **FGT segmentation: DSC = 0.86 (3D), 0.84 (2D)**
- **Blood vessels: DSC = 0.65 (3D), 0.53 (2D)**

Un elemento innovativo del lavoro è la segmentazione esplicita dei vasi, che ha mostrato impatto significativo nella stima della densità mammaria. In effetti, il volume dei vasi rappresentava in media il 5.7% del totale dei voxel etichettati come FGT o vasi. Trascurare que-

sto aspetto, come avveniva in studi precedenti, può portare a una sovrastima della densità mammaria.

Capitolo 3

Dataset: Duke Breast Cancer MRI

Contents

3.1 Composizione del Dataset

Il dataset utilizzato durante il tirocinio proviene dal database **Duke Breast Cancer MRI** [3], un archivio pubblico messo a disposizione dalla Duke University Medical Center e accessibile tramite il portale *The Cancer Imaging Archive (TCIA)*. Questo dataset è stato sviluppato per promuovere la ricerca nel campo della segmentazione automatica della mammella e contiene immagini di risonanza magnetica (MRI) acquisite da pazienti con diagnosi confermata di carcinoma mammario invasivo.

Ogni studio è composto da sequenze MRI in formato **DICOM**, ottenute in posizione prona utilizzando scanner da 1.5 T o 3.0 T (prodotti da GE Healthcare o Siemens). Le immagini sono acquisite in proiezione assiale, con una risoluzione spaziale variabile compresa tra $0.6 \times 0.6 \times 1.0$ mm^3 e $1.1 \times 1.1 \times 1.2$ mm^3, rendendole adatte all'elaborazione tridimensionale.

Le sequenze utilizzate nel progetto corrispondono alle immagini **T1-weighted fat-suppressed pre-contrast**, selezionate per l'elevato contrasto tra il tessuto adiposo e quello fibroghiandolare (FGT), senza l'interferenza di mezzi di contrasto. Questa scelta metodologica è coerente con quanto proposto da Lew et al. (2024), che hanno dimostrato l'efficacia di tale sequenza per la segmentazione automatica del FGT e dei vasi sanguigni.

3.1.1 Annotazioni e Ground Truth

Una caratteristica distintiva del dataset è la disponibilità di annotazioni manuali tridimensionali, elaborate da annotatori formati e successivamente validate da radiologi esperti in senologia. Ogni caso include segmentazioni voxel-wise delle seguenti strutture:

- **Seno (breast)**: delimitazione del tessuto mammario escludendo parete toracica, sternale e linfonodi;
- **Tessuto fibroghiandolare (FGT)**: zone a maggiore densità che costituiscono l'indicatore primario per la valutazione del rischio;
- **Vasi sanguigni**: rami dell'arteria mammaria interna e dell'arteria toracica laterale, spesso iperintensi anche in assenza di contrasto.

3.1.2 Preprocessing e Integrazione con MONAI

Durante la fase di preparazione, le immagini DICOM sono state convertite in un formato compatibile con **MONAI** (Medical Open Network for AI) [1], un framework open-source ottimizzato per l'elaborazione di dati medici in ambito deep learning.

La pipeline di preprocessing ha incluso le seguenti operazioni:

- il **caricamento** delle immagini e delle relative maschere;
- la **normalizzazione** dell'intensità per migliorare la stabilità del training;
- il **ridimensionamento** delle immagini a una risoluzione uniforme per l'input nei modelli di rete neurale;
- la conversione in **dizionari Python** contenenti l'immagine e la maschera;
- il **salvataggio** del dataset preprocessato su disco, pronto per essere utilizzato nel training.

3.1.3 Suddivisione del Dataset

Al fine di garantire una valutazione robusta e riproducibile dei modelli, l'intero dataset è stato suddiviso in tre sottoinsiemi:

- **Training set (80%)**: utilizzato per l'addestramento del modello, comprendente la maggior parte dei dati e delle variabilità cliniche;
- **Validation set (10%)**: impiegato durante l'addestramento per monitorare l'overfitting e ottimizzare gli iperparametri;
- **Test set (10%)**: utilizzato esclusivamente per la valutazione finale delle prestazioni del modello, senza interferenze in fase di sviluppo.

Capitolo 4

Configurazione Sperimentale

Contents

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 architetturali, 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 padding 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 dipendenze 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:

```yaml
# 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:

```yaml
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: monai.networks.nets.Unet
  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: monai.losses.DiceFocalLoss
  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:

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

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

POST_LABEL:
  - class: monai.transforms.AsDiscrete
    params:
      to_onehot: 2
```

4.2 Gestione del Logging

Per ogni esperimento è stato adottato un sistema di **logging automatico** organizzato in modo da raccogliere tutti gli output generati in una cartella dedicata, con l'obiettivo di garantire tracciabilità, riproducibilità e analisi approfondita dei risultati.

Ogni cartella di esperimento conteneva le seguenti informazioni:

- Un file `log.txt` contenente tutte le stampe console (`print`) prodotte durante il training. In questo file venivano riportati:
  - la loss di training e validazione per epoca;
  - le metriche di valutazione (es. Dice score);
  - eventuali messaggi diagnostici.
- Due grafici in formato immagine (`.png`):
  - il grafico della curva della loss per epoca;
  - il grafico della metrica per epoca.

- Il file del modello salvato (`.pth`), che rappresenta lo stato della rete al miglior checkpoint.
- Una sottocartella `predictions/` contenente, per ogni caso del test set, le immagini salvate slice per slice. Ogni immagine mostrava sovrapposte:
  - l'immagine originale della MRI dell'i-esima slice
  - la ground truth dell'i-esima slice
  - la predizione generata dal modello dell'i-esima slice

Questa struttura ha permesso non solo di monitorare l'andamento dell'addestramento, ma anche di effettuare confronti qualitativi tra modelli diversi e analizzare visivamente i punti di forza e debolezza della rete. Inoltre, la separazione per cartelle ha reso semplice la gestione di esperimenti multipli e il confronto sistematico tra diverse configurazioni di training.

Capitolo 5

Esperimenti

Bibliografia

[1] M. Jorge Cardoso, Wenqi Li, Tom Brown et al. *MONAI: An open-source framework for deep learning in healthcare*. arXiv preprint arXiv:2211.02701. 2022. URL: https://arxiv.org/abs/2211.02701.

[2] Jin Chen et al. *TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation*. arXiv preprint arXiv:2102.04306. 2021. URL: https://arxiv.org/abs/2102.04306.

[3] Kenneth Clark, Brian Vendt, Kirk Smith et al. *The Duke-Breast-Cancer-MRI Dataset*. The Cancer Imaging Archive (TCIA). https://doi.org/10.7937/TCIA.2019.4VAFYFM9. 2020.

[4] Christopher O. Lew et al. "A publicly available deep learning model and dataset for segmentation of breast, fibroglandular tissue, and vessels in breast MRI". In: *Scientific Reports* 14.1 (2024), p. 5383. DOI: 10.1038/s41598-024-54048-2. URL: https://doi.org/10.1038/s41598-024-54048-2.