



**UNIVERSITÀ DEGLI STUDI DI CAGLIARI**  
**FACOLTÀ DI SCIENZE**

Corso di Laurea Magistrale in Informatica

**Segmentazione di immagini  
istopatologiche con tecniche di  
apprendimento federato**

**Supervisors**  
Dott. Andrea Loddo

**Candidate**  
Marco Usai

ACADEMIC YEAR 2023/2024



# Abstract

La segmentazione di nuclei cellulari è un compito fondamentale nelle applicazioni di apprendimento automatico e profondo di immagini istopatologiche. Per questo motivo, i ricercatori hanno messo a disposizione un gran numero di dataset di immagini di risoluzioni differenti, utili per l’addestramento e il testing di modelli capaci di svolgere tale lavoro. Tuttavia, le immagini presentano caratteristiche diverse tra loro, perciò un singolo modello potrebbe non adattarsi a tutte le tipologie di immagini. L’apprendimento federato consente a più clienti di addestrare congiuntamente un modello di deep learning senza condividere i dati, siano essi immagini o relativi ai pazienti stessi. Grazie a queste tecniche sarà possibile addestrare modelli capaci di segmentare le immagini a prescindere dalle loro differenze di colorazione, forma, risoluzione o struttura, superando i problemi più comunemente noti come *domain shift*. I modelli locali addestrati verranno valutati su set di test provenienti dallo stesso dataset di training e su set di test esterni, per verificare la generalizzabilità dei modelli. Tra le tecniche di apprendimento federato, si sono considerate FedMix, FedAvg, FedST, FedRGD, confrontate con un addestramento esclusivamente locale, denominato Local Learning. Tali tecniche sfruttano diverse strategie di supervisione e aggregazione per l’addestramento collaborativo dei modelli. I risultati sperimentali ottenuti su più dataset pubblici mostrano che l’approccio proposto permette un miglioramento rispetto ai metodi allo stato dell’arte di un ampio margine. Questa ricerca offre un contributo significativo permettendo di migliorare la segmentazione dei nuclei cellulari da immagini istopatologiche in un contesto di generalizzabilità del dominio.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Contesto e motivazioni . . . . .	1
1.2	Obiettivi della ricerca . . . . .	2
<b>2</b>	<b>Stato dell'arte</b>	<b>5</b>
2.1	Stato dell'arte . . . . .	5
2.1.1	Panoramica sull'imaging istopatologico . . . . .	5
2.1.2	Applicazioni di segmentazione nell'istopatologia . . . . .	6
2.1.3	Deep Learning . . . . .	7
2.1.4	Federated Learning . . . . .	8
<b>3</b>	<b>Materiali e metodi</b>	<b>11</b>
3.1	Dataset . . . . .	11
3.1.1	MICCAI . . . . .	11
3.1.2	BNS . . . . .	12
3.1.3	MoNuSeg . . . . .	13
3.1.4	TNBC . . . . .	14
3.1.5	MoNuSAC . . . . .	15
3.2	Tipologie di apprendimento . . . . .	15
3.2.1	Apprendimento supervisionato . . . . .	16
3.2.2	Apprendimento semi-supervisionato . . . . .	16
3.2.3	Apprendimento di supervisione mista . . . . .	16
3.3	Tecniche di apprendimento impiegate . . . . .	17
3.3.1	FedAvg . . . . .	17
3.3.2	FedMix . . . . .	18
3.3.3	FedST . . . . .	19
3.3.4	FedRGD . . . . .	20

3.3.5	Local Learning . . . . .	21
3.4	Caratteristiche e configurazioni del training . . . . .	21
3.4.1	Split in immagini di train, val e test . . . . .	21
3.4.2	Architettura U-Net . . . . .	23
3.4.3	Parametri di configurazione . . . . .	23
3.4.4	Ottimizzatore Adam . . . . .	24
3.4.5	Loss Function . . . . .	25
3.4.6	Processo di data augmentation . . . . .	25
3.5	Misure di performance dei modelli . . . . .	27
<b>4</b>	<b>Risultati sperimentali</b>	<b>33</b>
4.1	Risultati in-dataset . . . . .	33
4.2	Risultati su dataset out-of-distribution . . . . .	35
4.3	Risultati apprendimento incrementale in-dataset . . . . .	36
4.4	Risultati apprendimento incrementale su dataset OOD . . . . .	37
4.5	Sperimentazione cross-dataset . . . . .	39
4.6	Comparazione conclusiva . . . . .	41
4.7	Discussione . . . . .	43
<b>5</b>	<b>Conclusioni e sviluppi futuri</b>	<b>45</b>
<b>References</b>		<b>53</b>

# Capitolo 1

## Introduzione

L'analisi delle immagini istopatologiche rappresenta un aspetto cruciale nella diagnosi e nel trattamento delle malattie, richiedendo una notevole precisione e un impegno significativo da parte dei medici e dei ricercatori. Con il progresso delle tecnologie di apprendimento automatico e profondo, si sono aperte nuove opportunità per automatizzare e migliorare queste analisi. Questo capitolo introduce il contesto e le motivazioni del lavoro in Sezione 1.1, delineando gli obiettivi della ricerca e spiegando come l'apprendimento federato possa offrire soluzioni avanzate per la segmentazione delle immagini istopatologiche in Sezione 1.2.

### 1.1 Contesto e motivazioni

Nel contesto dell'analisi delle immagini istopatologiche, la segmentazione dei nuclei cellulari rappresenta un'applicazione cruciale della computer vision, consentendo di identificare e analizzare dettagli morfologici e tessiturali fondamentali per la diagnosi delle patologie [14, 23]. Tuttavia, mentre le tecnologie di apprendimento automatico e profondo hanno rivoluzionato questo campo, rimane una sfida significativa: il *domain shift* [26].

Il domain shift si riferisce alla discrepanza tra le distribuzioni dei dati di addestramento e di test. Nell'ambito delle immagini istopatologiche, questo fenomeno si manifesta quando modelli addestrati su un determinato dataset non generalizzano efficacemente su nuovi dati provenienti da altre fonti o con caratteristiche diverse. Le variazioni nelle condizioni di acquisizione delle immagini,

come colorazione, risoluzione e condizioni di illuminazione, possono influenzare pesantemente le prestazioni di un modello di segmentazione [2].

Affrontare il domain shift è cruciale per migliorare l'affidabilità e l'applicabilità dei modelli di segmentazione nelle applicazioni cliniche. Strategie come l'apprendimento federato emergono come potenziali soluzioni, consentendo l'addestramento distribuito di modelli su dati locali mantenendo la privacy dei dati dei pazienti. Questo approccio non solo può migliorare la robustezza del modello su diverse fonti di dati, ma anche ridurre il rischio di overfitting e migliorare la generalizzazione su nuovi dati.

Nel prosieguo della tesi, sarà essenziale esplorare come l'apprendimento federato possa essere implementato efficacemente per affrontare il domain shift nelle applicazioni di segmentazione delle immagini istopatologiche. Questo includerà l'analisi delle metodologie di addestramento federato, la valutazione delle prestazioni in scenari realistici e l'implementazione di strategie per mitigare le disparità di dominio nei dati clinici.

## 1.2 Obiettivi della ricerca

Per affrontare queste sfide, questa ricerca si è concentrata sull'impiego di tecniche di apprendimento federato per la segmentazione delle immagini istopatologiche. L'apprendimento federato ha permesso di addestrare modelli di deep learning su dati distribuiti tra più client, senza la necessità di centralizzare i dati. Questa metodologia ha consentito di sfruttare un'ampia varietà di dati di training, migliorando la capacità del modello di generalizzare su diverse tipologie di immagini.

In questo lavoro, sono stati sviluppati e addestrati vari modelli di apprendimento federato utilizzando dataset pubblicamente disponibili di immagini istopatologiche. I modelli sono stati testati sia su set di test provenienti dagli stessi dataset di training, sia su set di test esterni, per valutare la loro capacità di generalizzazione. Attraverso sperimentazioni e confronti tra i vari modelli, si è cercato di determinare quali tecniche di apprendimento federato offrano le migliori performance in termini di accuratezza, precisione ed efficienza nella segmentazione delle immagini.

L'obiettivo principale di questo lavoro è stato dimostrare che l'apprendimento federato può superare le limitazioni dei modelli tradizionali, offrendo una soluzione robusta e scalabile per la segmentazione delle immagini istopatologiche. I risultati di questa ricerca hanno potuto fornire un contributo significativo al campo della diagnostica medica, migliorando la qualità delle diagnosi e riducendo il tempo necessario per analizzare le immagini.

Attraverso un'analisi approfondita dei risultati sperimentali, si è identificato quali tecniche sono state più efficaci e si sono proposti miglioramenti per futuri sviluppi. Questo lavoro si è posto come punto di riferimento per futuri studi sull'applicazione dell'apprendimento federato in ambito medico, promuovendo l'adozione di queste tecniche innovative nella pratica clinica.



# **Capitolo 2**

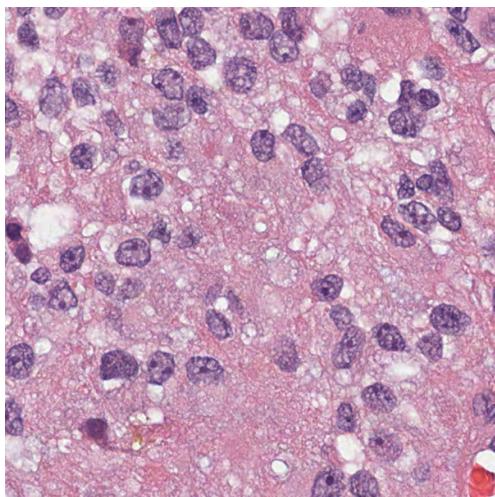
## **Stato dell'arte**

### **2.1 Stato dell'arte**

In questo capitolo si fornirà una panoramica generale sull'analisi delle immagini istopatologiche in Sezione 2.1.1; elencando le principali applicazioni per effettuare le diagnosi in Sezione 2.1.2 e mostrando le tecniche di apprendimento utilizzate per generare modelli capaci di eseguire il task di segmentazione in maniera automatica ed efficiente. In particolare, si discuteranno l'apprendimento profondo in Sezione 2.1.3 e l'apprendimento federato in Sezione 2.1.4.

#### **2.1.1 Panoramica sull'imaging istopatologico**

L'istopatologia è lo studio dei segni di una malattia attraverso l'esame al microscopio di un campione biotecnico o chirurgico (biopsia) trattato e fissato su vetrini [24]. L'imaging istopatologico è una tecnica diagnostica essenziale in medicina, utilizzata per l'analisi microscopica dei tessuti. Questa tecnica consente di osservare dettagli morfologici e strutturali delle cellule e dei tessuti, permettendo ai patologi di identificare segni di malattie, come il cancro, infezioni o altre condizioni patologiche. Le immagini istopatologiche sono ottenute mediante microscopi ottici e possono essere colorate utilizzando diverse tecniche di colorazione, per evidenziare specifiche caratteristiche cellulari e tissutali [7].



**Figura 2.1:** Esempio di immagine istopatologica catturata al microscopio [15].

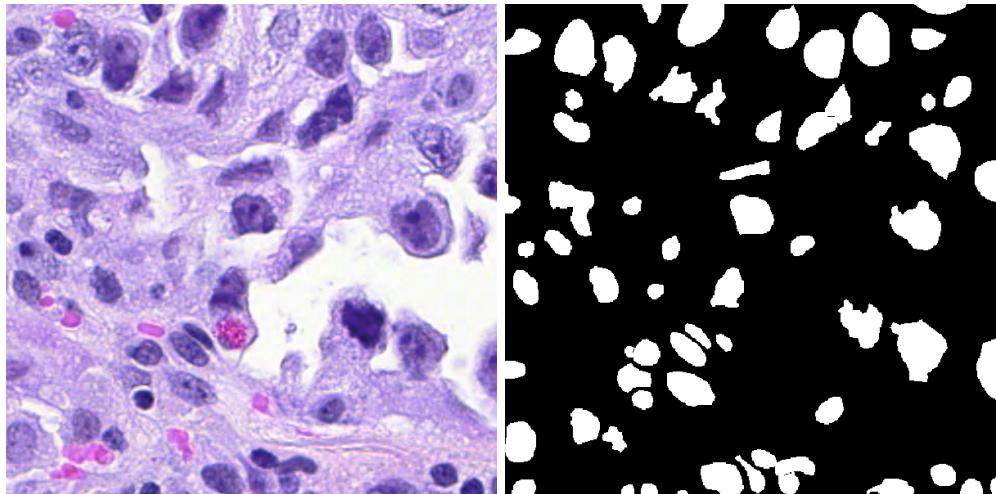
Le immagini risultanti presentano una vasta gamma di variazioni in termini di colorazione, risoluzione e struttura, a seconda della tecnica di preparazione del campione e del tipo di tessuto esaminato. Queste variazioni rendono l’analisi automatica delle immagini una sfida significativa, richiedendo modelli di apprendimento automatico capaci di generalizzare su diverse tipologie di dati. Un esempio di immagine istopatologica è riportato in Figura 2.1.

### 2.1.2 Applicazioni di segmentazione nell’istopatologia

La segmentazione di nuclei cellulari dalle immagini istopatologiche è cruciale per l’analisi accurata delle caratteristiche cellulari e tissutali. Questo processo non solo consente di estrarre il numero di regioni e la distribuzione delle cellule, ma anche di ricavare informazioni come la forma e la struttura morfologica. Questi dati permettono ai ricercatori di identificare regioni anomale che possono rivelare informazioni fondamentali nello studio delle patologie.

Com’è possibile notare dagli esempi riportati in Figura 2.2, i vari dataset utilizzati per il training o il test contengono una moltitudine di coppie immagine-ground truth, in modo che potranno essere utilizzate per far capire al modello come dovrà operare su ogni immagine che gli verrà data in input. Dovrà dunque essere in grado di estrarre i nuclei cellulari effettuando le predizioni corrette delle etichette. Maggiormente le predizioni delle etichette del modello saranno simili alle etichette reali, più il modello sarà considerato efficiente. Un esempio

di coppie immagine-ground truth relativi a due dataset differenti è riportato in Figura 2.3.



**Figura 2.2:** Esempio di coppia immagine-ground truth per la segmentazione di immagini istopatologiche [15].

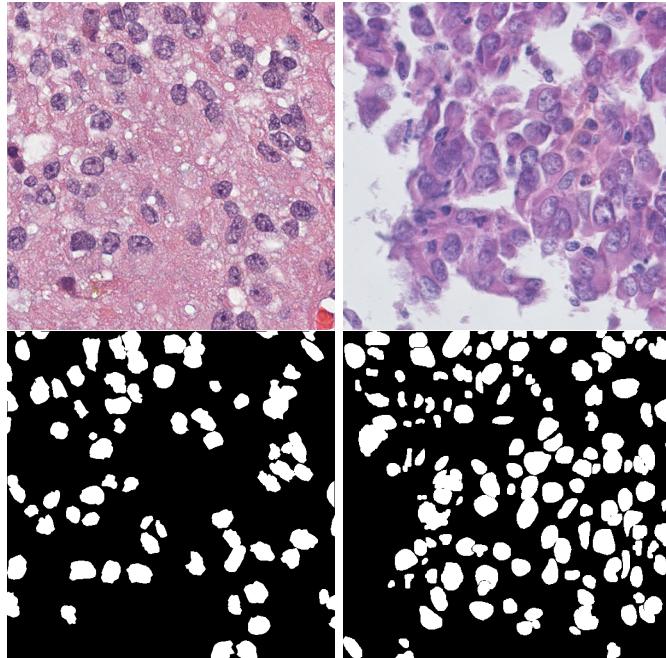
### 2.1.3 Deep Learning

Il deep learning è una disciplina dell'intelligenza artificiale che utilizza reti neurali artificiali profonde per apprendere da dati non strutturati o complessi. Questi modelli sono caratterizzati da diversi strati di neuroni, ciascuno dei quali esegue trasformazioni complesse sui dati in ingresso per identificare modelli, pattern e rappresentazioni significative.

Le Convolutional Neural Networks (CNN) sono una categoria specializzata di reti neurali profonde progettate principalmente per l'analisi di dati strutturati, soprattutto immagini o video. Le CNN sfruttano operazioni di convoluzione per estrarre automaticamente caratteristiche gerarchiche dai dati, imparando progressivamente a riconoscere pattern sempre più complessi man mano che procedono attraverso i vari strati di neuroni della rete. Questa metodologia rende le CNN particolarmente efficaci nell'analisi delle immagini, poiché possono catturare dettagli locali e relazioni spaziali tra i pixel [12].

Nel contesto dell'analisi delle immagini istopatologiche, le CNN sono ampiamente utilizzate per compiti come la segmentazione dei nuclei cellulari, la classificazione delle anomalie tessutali e altre attività di analisi morfologica. La capacità

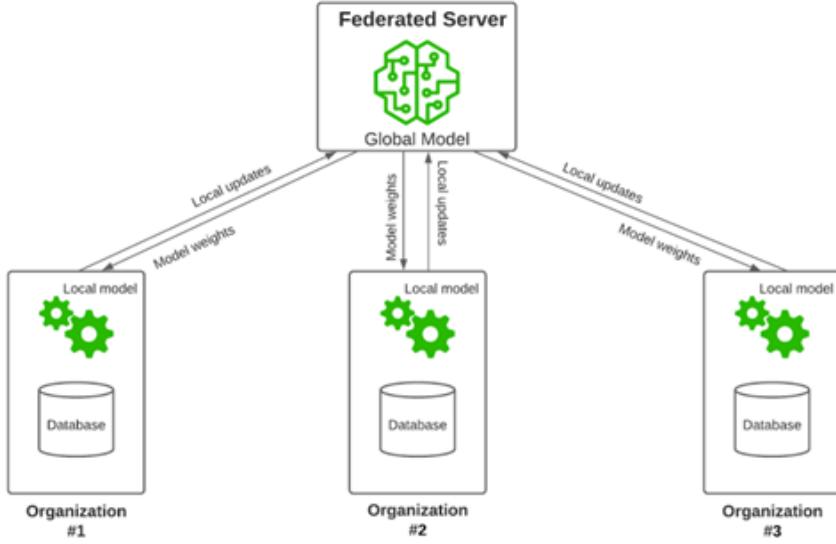
delle CNN di apprendere rappresentazioni discriminative direttamente dai dati grezzi ha rivoluzionato molti settori, inclusa la diagnostica medica, dove la precisione e l'efficienza sono cruciali. In seguito, verrà illustrato il funzionamento della CNN U-Net, utilizzata per il task di segmentazione.



**Figura 2.3:** Esempio di immagine e relativa ground truth di due dataset differenti: MICCAI e BNS [15].

#### 2.1.4 Federated Learning

Il federated learning (FL) è una metodologia di apprendimento che prevede l'aggregazione di più modelli in uno singolo, permettendo di generalizzare su più dati in modo efficace. La sua importanza sta nel fatto che, seppur i modelli siano incentrati sullo svolgimento di un particolare task e meno precisi in altri, l'apprendimento federato può generare un modello in grado di compiere i vari task in modo più efficiente. Nel caso dell'istopatologia, le immagini sono state raccolte esaminando le varie cellule al microscopio, da vari ospedali. Le informazioni che sono state raccolte e le immagini che sono state prodotte sono quindi diverse da ospedale a ospedale, dunque l'impiego di questa metodologia di apprendimento si può rivelare molto utile per ovviare a queste differenze di morfologia e caratteristiche [10]. Le problematiche relative alla privacy dei pazienti sono cruciali



**Figura 2.4:** Organizzazione dell'apprendimento federato<sup>1</sup>.

in ambito medico e il FL offre una soluzione promettente per affrontarle. Mentre i dati localmente e condividendo solo le informazioni necessarie tramite aggiornamenti di modello, si riduce il rischio di esposizione di dati sensibili. Ciò è particolarmente importante in settori altamente regolamentati come la sanità, dove la conformità alle normative sulla privacy come il GDPR in Europa o l'HIPAA negli Stati Uniti è fondamentale [6]. L'impiego del FL nell'analisi delle immagini istopatologiche non solo può migliorare la precisione diagnostica e la capacità predittiva, ma anche promuovere la collaborazione tra istituti medici senza compromettere la sicurezza e la riservatezza dei dati dei pazienti. In sostanza, anziché inviare dati grezzi a un server centrale per l'addestramento, il FL consente ai dispositivi di collaborare localmente per addestrare un modello globale. Il processo di addestramento inizia con la distribuzione di un modello di base ai dispositivi locali. Come osservabile nella Figura 2.4, questi dispositivi addestrano il modello utilizzando i loro dati locali e quindi inviano solo i pesi aggiornati, anziché i dati grezzi, a un server centrale. Il server aggrega quindi i pesi da tutti i dispositivi e aggiorna il modello globale. Questo ciclo di aggiornamento può ripetersi iterativamente fino a quando il modello raggiunge una prestazione accettabile o soddisfa i criteri di convergenza. Il FL è particolarmente utile in

---

<sup>1</sup>Federated Learning: Predictive Model Without Data Sharing

scenari in cui i dati sono sensibili o distribuiti in modo eterogeneo e non possono essere facilmente condivisi centralmente.

# **Capitolo 3**

## **Materiali e metodi**

In questo capitolo verranno fornite informazioni per quanto riguarda i dataset utilizzati per il training e il testing dei modelli di apprendimento automatico in Sezione 3.1; quali tipologie di apprendimento sono state impiegate per generarli in Sezione 3.2, includendo descrizioni dettagliate sulle varie tecniche utilizzate per addestrare i modelli in Sezione 3.3 e sulle varie configurazioni del training e dei modelli in Sezione 3.4. Infine, verrà mostrata una panoramica sulla valutazione finale dei modelli in Sezione 3.5, includendo le misure di performance e i grafici considerati per comprenderne al meglio le prestazioni.

### **3.1 Dataset**

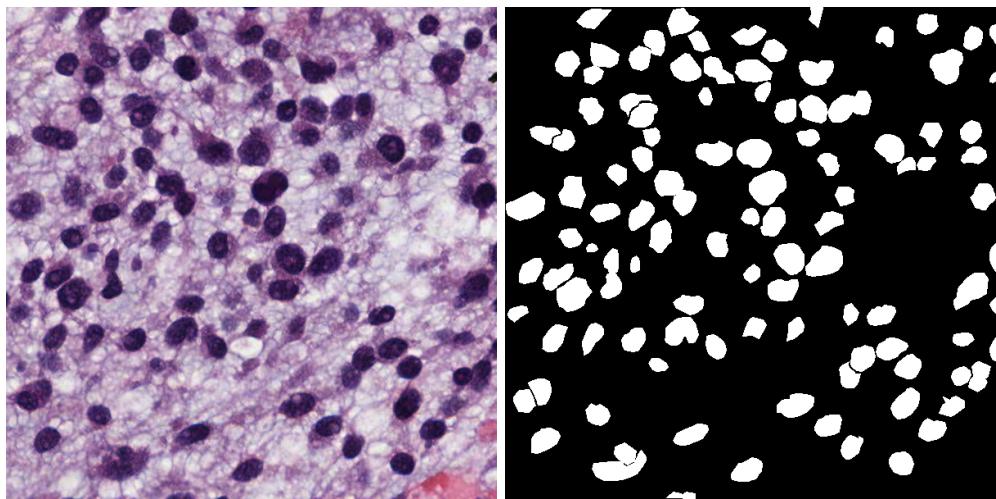
#### **3.1.1 MICCAI**

Il primo insieme di immagini utilizzato per addestrare e testare i modelli di apprendimento federato è il dataset MICCAI. Questo dataset, reso pubblico per la Challenge di Digital Pathology MICCAI 2017<sup>1</sup>, comprende 32 immagini quadrate annotate, con dimensioni di  $500 \times 500$  o  $600 \times 600$  pixel. Le immagini sono state estratte manualmente da Whole Slide Images (WSIs) colorate con H&E (ematosilina ed eosina), una tecnica comune in istopatologia per migliorare il contrasto dei tessuti [33]. Le immagini rappresentano quattro tipi di cancro: glioblastoma (GBM), glioma di basso grado (LGG), carcinoma a cellule squamose della testa e del collo (HNSC) e carcinoma a cellule squamose dei polmoni (LUSC). Cia-

---

<sup>1</sup>MICCAI 2017 Digital Pathology Challenge

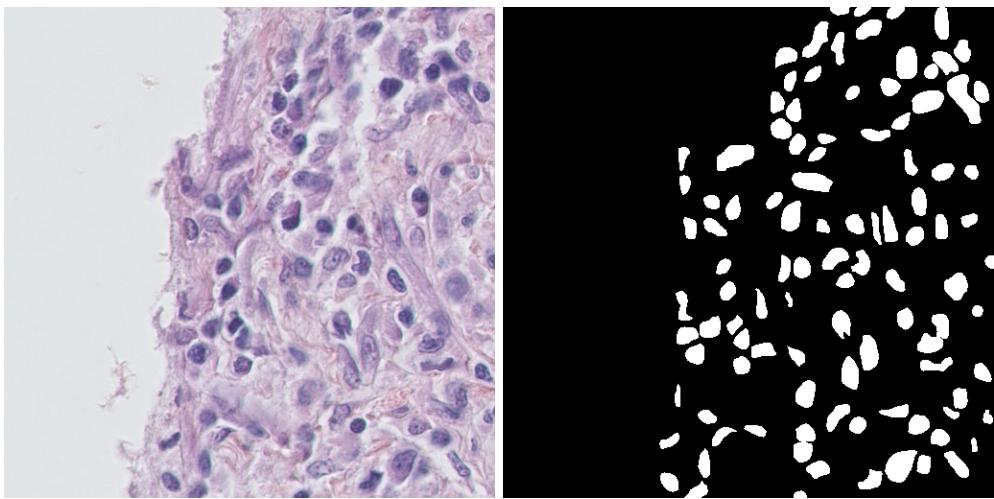
scuna immagine è accompagnata dalla sua ground truth corrispondente, estratta manualmente da patologi esperti. Un esempio è riportato in Figura 3.1. Entrambe le immagini ed ground truth sono fornite in formato PNG. Per agevolare il riconoscimento da parte dei modelli, le immagini sono RGB e le ground truth sono state convertite in bianco e nero. Inoltre, per aumentare la quantità di dati di addestramento, è stato applicato un processo di data augmentation.



**Figura 3.1:** Esempio di coppia immagine-ground truth MICCAI [15].

### 3.1.2 BNS

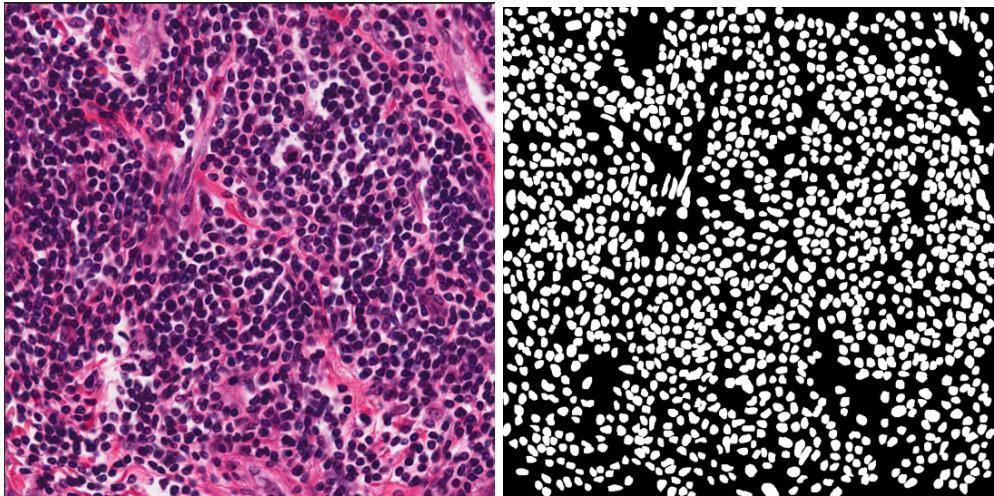
Il secondo dataset utilizzato in questo lavoro è BNS, presentato da Naylor et al. [19]. Questo dataset comprende 33 immagini quadrate annotate, anch'esse estratte da WSIs colorate con H&E, con dimensioni di  $512 \times 512$  pixel. Le immagini rappresentano diverse patologie e sono state annotate manualmente da patologi esperti. Come per il dataset MICCAI, sia le immagini che le ground truth sono fornite in formato PNG e sono state convertite in bianco e nero per migliorare il riconoscimento da parte dei modelli. Un esempio è riportato in Figura 3.2. Anche per questo dataset è stato applicato un processo di data augmentation [16].



**Figura 3.2:** Esempio di coppia immagine-ground truth BNS [15]

### 3.1.3 MoNuSeg

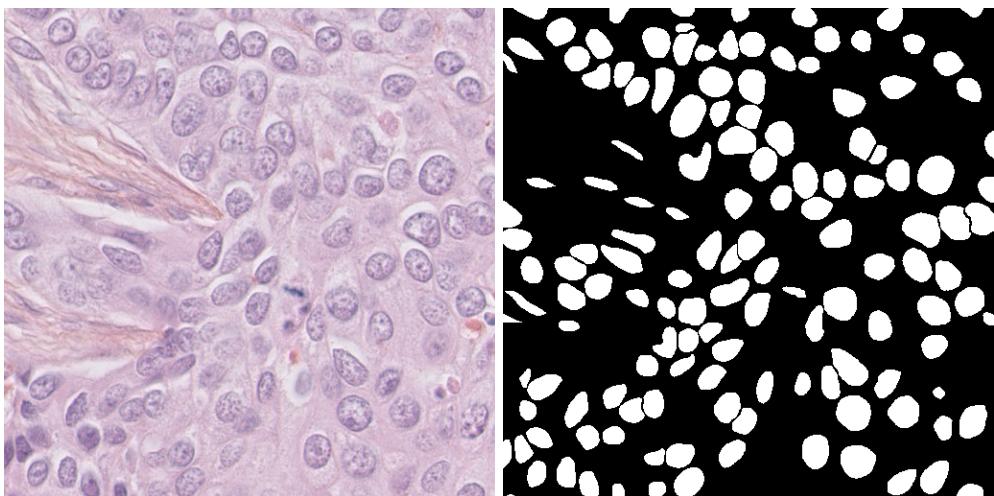
Il terzo dataset utilizzato è MoNuSeg, ottenuto annotando attentamente immagini di tessuti di diversi pazienti con tumori di vari organi, diagnosticati in diversi ospedali. Questo dataset è stato creato scaricando immagini di tessuti colorati con H&E, catturate a 40x di ingrandimento dall'archivio TCGA. Le immagini rappresentano una vasta diversità di tumori e di tessuti. Per uniformare le varie tonalità di colorazione e migliorare la qualità delle immagini, sono state applicate delle tecniche di pre-elaborazione. Le immagini, fornite in formato PNG con una risoluzione di  $1000 \times 1000$  pixel, sono state convertite in scala di grigi per semplificare il processo di segmentazione dei nuclei. Un esempio è riportato in Figura 3.3. Anche per questo dataset è stato applicato un processo di data augmentation [13].



**Figura 3.3:** Esempio di coppia immagine-ground truth MoNuSeg. Fonte: [13].

### 3.1.4 TNBC

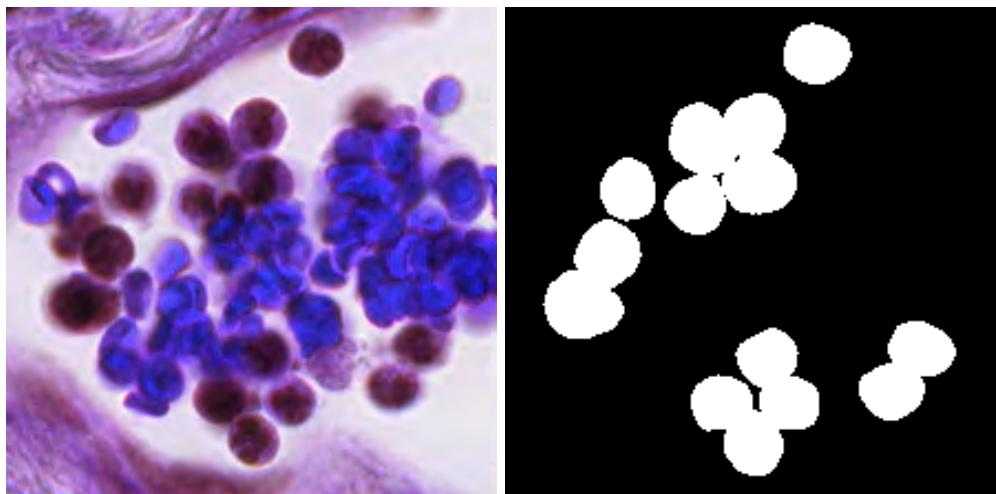
Il dataset TNBC (Triple-negative breast cancer) è stato utilizzato esclusivamente per il testing dei modelli. Questo dataset comprende 50 immagini annotate, con dimensioni di  $512 \times 512$  pixel, prelevate da 11 pazienti diversi. Le immagini mostrano una varietà di cellule, tra cui cellule normali del seno, cellule carcinomatose invasive, fibroblasti, cellule endoteliali e altre [21]. Le immagini e le relative ground truth sono state annotate da esperti in patologia, con un'attenzione particolare alla precisione e alla completezza delle annotazioni [20]. Un esempio è riportato in Figura 3.4.



**Figura 3.4:** Esempio di coppia immagine-ground truth TNBC [9].

### 3.1.5 MoNuSAC

Infine, il dataset MoNuSAC contiene immagini di tessuti provenienti da quattro organi diversi: polmone, prostata, rene e seno. Questo dataset include sia immagini che maschere binarie delle regioni ambigue, che non sono state considerate nel calcolo delle misure di performance durante la classificazione delle submission. Le immagini e le relative ground truth sono fornite in formato .svs e .tif, processate con una funzione della libreria di Python "OpenSlide", insieme a 31.000 annotazioni dei confini nucleari in file .xml. Questo dataset è stato utilizzato per valutare le prestazioni dei modelli in contesti multi-organici, consentendo di testare la generalizzazione dei modelli su diverse tipologie di tessuti. Un esempio è riportato in Figura 3.5. Come per gli altri dataset, anche per MoNuSAC è stato applicato un processo di data augmentation per aumentare la diversità dei dati di addestramento [28].



**Figura 3.5:** Esempio di coppia immagine-ground truth MoNuSAC [28].

## 3.2 Tipologie di apprendimento

Nell'ambito dell'intelligenza artificiale e del machine learning, le tipologie di apprendimento delineano il modo in cui i modelli elaborano i dati e migliorano le loro prestazioni. Tra le principali tipologie vi sono l'apprendimento supervisionato, semi-supervisionato e di supervisione mista. Queste metodologie svolgono un ruolo fondamentale nella segmentazione dei dataset utilizzati per l'addestra-

mento dei modelli di federated learning. Ad esempio, nell'apprendimento supervisionato, ogni immagine viene associata a un ground truth corrispondente, che guida il modello durante il processo di apprendimento. Nel caso dell'apprendimento semi-supervisionato, alcune immagini possono non essere etichettate, il che può influenzare la capacità del modello di generalizzare su nuovi dati. L'apprendimento di supervisione mista combina elementi di entrambi, consentendo al modello di sfruttare sia le informazioni etichettate che non etichettate per migliorare le sue capacità predittive.

### **3.2.1 Apprendimento supervisionato**

L'apprendimento supervisionato è una tecnica in cui il modello è addestrato su un insieme di dati che contiene esempi di input e le rispettive ground truth corrette. Questo tipo di apprendimento è ampiamente utilizzato in problemi di classificazione e regressione. Nel contesto della segmentazione dei dataset istopatologici, l'apprendimento supervisionato è essenziale per associare correttamente le immagini alle relative ground truth. Ad esempio, un modello potrebbe essere addestrato su un dataset di immagini istopatologiche annotate manualmente con le regioni di interesse, come nuclei cellulari o tessuti patologici, per identificare automaticamente queste regioni nei nuovi campioni [11].

### **3.2.2 Apprendimento semi-supervisionato**

L'apprendimento semi-supervisionato implica l'addestramento del modello su un insieme di dati contenente sia esempi con ground truth che esempi senza ground truth. Questo tipo di apprendimento è utile quando è costoso ottenere un gran numero di ground truth manuali. Nella segmentazione dei dataset istopatologici, ad esempio, un modello potrebbe essere addestrato su un insieme limitato di dati con annotazioni esplicite, ma potrebbe anche utilizzare tecniche di clustering o altri metodi non supervisionati per migliorare la segmentazione di nuove immagini, riducendo così il carico di lavoro di annotazione umana [1].

### **3.2.3 Apprendimento di supervisione mista**

L'apprendimento di supervisione mista combina elementi dell'apprendimento supervisionato e non supervisionato, addestrando il modello su un insieme di dati

contenente una combinazione di esempi con e senza ground truth. Questo tipo di approccio è particolarmente utile quando si desidera sfruttare al massimo tutte le informazioni disponibili nel dataset. Nella segmentazione dei dataset istopatologici, un modello di supervisione mista potrebbe essere addestrato su un insieme di immagini che comprende alcuni esempi annotati manualmente e altri esempi non annotati. Questo consentirebbe al modello di apprendere dalle annotazioni disponibili e di esplorare automaticamente estrarre altre caratteristiche rilevanti dai dati non annotati, migliorando così la precisione complessiva della segmentazione [30].

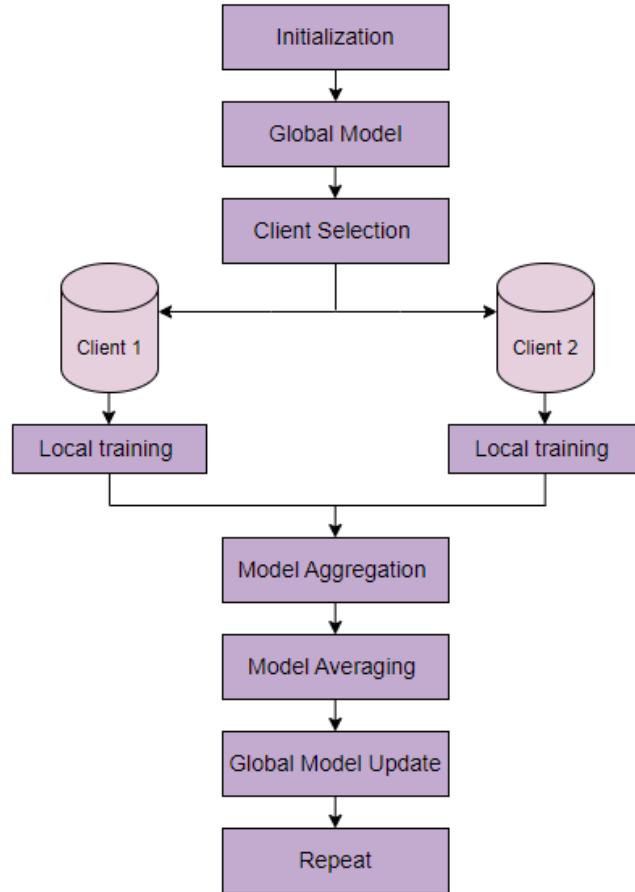
### 3.3 Tecniche di apprendimento impiegate

#### 3.3.1 FedAvg

La media federata (FedAvg) è una tecnica innovativa e fondamentale nell'ambito dell'apprendimento federato, utilizzata per addestrare modelli di machine learning distribuiti su una vasta rete di dispositivi [18]. Questo approccio affronta efficacemente le sfide legate alla privacy e alla sicurezza dei dati, mantenendo i dati sensibili decentralizzati sui dispositivi periferici senza la necessità di condividerli centralmente. In un contesto tradizionale di apprendimento automatico centralizzato, i dati vengono raccolti e archiviati in un server centrale, dove viene addestrato un unico modello su questi dati consolidati. Tuttavia, questo approccio presenta limitazioni significative in termini di privacy, sicurezza e scalabilità. Al contrario, FedAvg, come visibile in Figura 3.6, adotta un approccio decentralizzato, distribuendo l'addestramento del modello sui dispositivi periferici, consentendo ai dati di rimanere locali e protetti.

L'addestramento del modello in FedAVG avviene mediante una procedura di aggregazione dei gradienti calcolati sui dati locali dei dispositivi. Questa procedura consente di aggiornare il modello globalmente, sfruttando le informazioni distribuite su tutta la rete senza la necessità di condividere direttamente i dati grezzi. Inoltre, FedAvg incorpora meccanismi di sicurezza avanzati per garantire l'integrità e la riservatezza dei dati durante il processo di addestramento federato. Questo approccio offre numerosi vantaggi, tra cui una maggiore sicurezza e privacy dei dati, una migliore scalabilità e una maggiore efficienza computazionale. Inoltre, FedAvg rappresenta un passo avanti significativo verso l'implementazione.

tazione di sistemi di apprendimento automatico su larga scala, consentendo una collaborazione distribuita e sicura tra dispositivi intelligenti.



**Figura 3.6:** Struttura di un sistema di FedAvg<sup>2</sup>.

### 3.3.2 FedMix

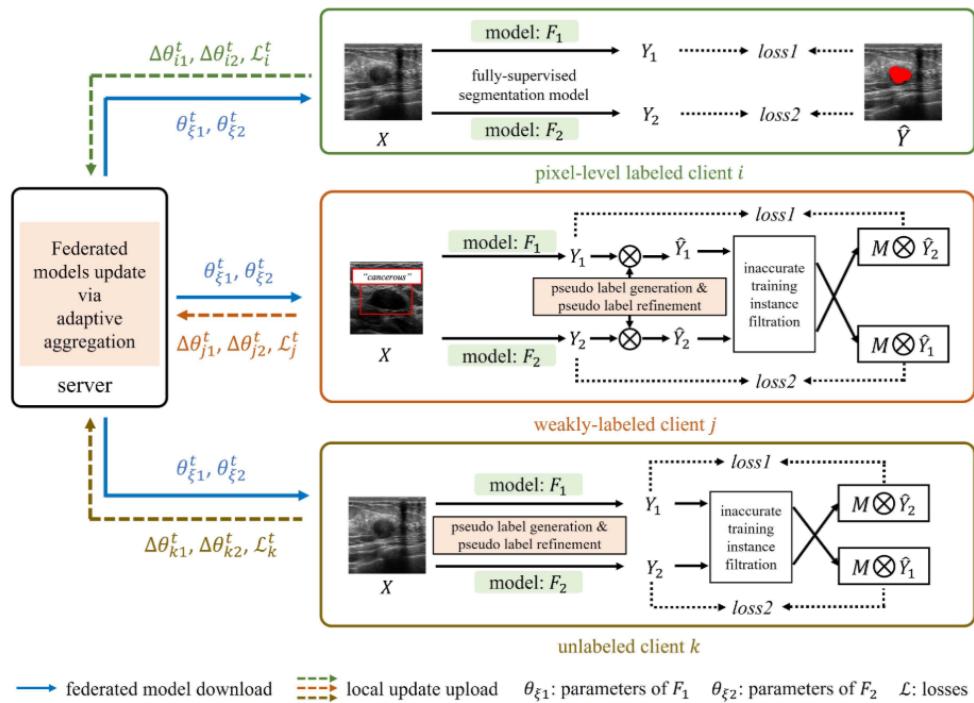
La tecnica di apprendimento federato misto (FedMix) sfrutta appieno i dati etichettati in qualsiasi forma per addestrare un modello di FL unificato per la segmentazione di immagini mediche. Si garantiscono contributi positivi dagli aggiornamenti del modello locale di ciascun client attraverso un processo in due fasi. In primo luogo, in assenza di ground truth a livello di pixel, FedMix utilizza efficacemente l'addestramento consistente per estrarre informazioni utili dalle

---

<sup>2</sup>What is Federated Averaging (FedAvg)?

immagini non etichettate così come dalle immagini debolmente etichettate (ossia ground truth di classe a livello di immagine e ground truth di bounding box) per produrre e selezionare pseudo-ground truth di alta qualità utilizzate per gli aggiornamenti del modello locale. Attraverso un processo iterativo, l'accuratezza delle pseudo-ground truth selezionate migliora gradualmente, portando a una migliore performance del modello locale.

Successivamente, per stimare meglio il potenziale contributo di ciascun client, viene utilizzata come misura di performance una funzione di perdita per regolare adattivamente il peso di aggregazione di ciascun client nell'aggiornamento del modello federato, il che è vitale per gestire le variazioni tra client nella disponibilità di ground truth. Questa tecnica ricade nella tipologia di apprendimento di supervisione mista e la sua struttura è riportata in Figura 3.7.



**Figura 3.7:** Struttura di un sistema di FedMix [30].

### 3.3.3 FedST

La tecnica di self-training federato (FedST) sfrutta sia dati etichettati a livello di pixel che non etichettati [4] per l'addestramento federato. FedST è stato scelto

per il confronto poiché richiede solo la condivisione dei parametri del modello aggiornati localmente, senza ulteriori informazioni condivise tra i client.

L'approccio di self-training consente ai client di migliorare la precisione del modello utilizzando iterativamente i dati etichettati localmente acquisiti durante l'addestramento. Ciò significa che il modello migliora continuamente la sua capacità di generalizzazione senza dipendere da un flusso costante di dati etichettati centralmente [32].

FedST può essere particolarmente utile in scenari in cui i dati etichettati sono limitati o costosi da ottenere, consentendo ai client di sfruttare appieno i dati locali non etichettati per migliorare le prestazioni del modello federato. Tuttavia, è importante considerare i rischi associati all'auto-annotazione, come la possibilità di introdurre errori o bias nel processo di addestramento.

### 3.3.4 FedRGD

La tecnica di apprendimento federato tramite riduzione della diversità del gradiente (FedRGD) mira a minimizzare la diversità del gradiente [17] tra i modelli dei clienti sostituendo la normalizzazione a batch con la normalizzazione a gruppi e utilizzando un nuovo approccio di aggregazione del modello. Questo metodo è particolarmente efficace nel gestire variazioni significative tra i client nella disponibilità di dati etichettati.

FedRGD si concentra sulla regolarizzazione del processo di aggiornamento del modello federato per garantire una maggiore coerenza tra i modelli dei clienti. Questo è fondamentale per garantire una convergenza stabile e uniforme del modello federato su tutti i client, specialmente quando i dati disponibili tra i client presentano variazioni significative in termini di distribuzione o quantità.

L'adozione della normalizzazione a gruppi e di un nuovo metodo di aggregazione del modello consente a FedRGD di gestire in modo più efficace la variazione dei dati tra i client, migliorando così la robustezza e le prestazioni del modello federato su una vasta gamma di scenari di distribuzione dei dati. Tuttavia, è importante considerare l'impatto computazionale aggiuntivo di questi approcci più complessi sull'efficienza complessiva del sistema federato [34].

### **3.3.5 Local Learning**

Il Local Learning (LL) consiste nell’addestrare un modello localmente. Siccome non fa parte della categoria delle tecniche di apprendimento federato, è stato impiegato come termine di comparazione per i restanti modelli. Ogni client addestra una propria CNN basata esclusivamente sui propri dati locali. Questa metodologia consente ai client di adattare il modello alle caratteristiche specifiche dei propri dati, senza la necessità di condividere informazioni sensibili o di dipendere da un server centrale per l’addestramento.

Nel contesto del LL, ogni client utilizza i propri dati locali per addestrare un modello CNN, sfruttando le informazioni contenute nei propri dati per migliorare la precisione e la generalizzazione del modello. Questo approccio può essere particolarmente vantaggioso in scenari in cui i dati hanno una distribuzione geografica o una natura altamente eterogenea, poiché consente una personalizzazione accurata del modello su scala locale.

Tuttavia, il LL può presentare alcune sfide, come la necessità di gestire la variazione dei dati tra i client e la possibile divergenza dei modelli locali nel tempo. È fondamentale implementare meccanismi di coordinamento e aggregazione dei modelli locali per garantire la coerenza e la convergenza del modello federato nel tempo.

Nonostante queste sfide, il LL offre numerosi vantaggi, tra cui una maggiore privacy e sicurezza dei dati, una maggiore flessibilità nell’adattamento del modello alle specifiche esigenze dei client e una riduzione del carico computazionale sul server centrale. Questo lo rende un’opzione attraente in molte applicazioni di machine learning distribuito, specialmente in contesti in cui la privacy e la sicurezza dei dati sono di primaria importanza.

## **3.4 Caratteristiche e configurazioni del training**

### **3.4.1 Split in immagini di train, val e test**

Per eseguire l’addestramento dei vari modelli, sono stati impiegati due split delle immagini del dataset. Il primo prevede una suddivisione 8:2 per immagini di training e validation, considerando esclusivamente le immagini originali non augmentate. Una volta che il primo split è andato a buon fine, sono stati aggiunti

i campioni aumentati al data loader contenente le immagini di training. Una volta che l’addestramento ha avuto inizio, per ogni epoca, i modelli vengono addestrati sullo 0.8 di immagini originali e sul 100% di immagini aumentate, e successivamente validati su quello 0.2 iniziale di immagini di validation. Il codice utilizzato per lo split è mostrato in Figura 3.8. Quando il training si è concluso, sono stati testati i modelli su campioni di test sia interni agli stessi dataset utilizzati per l’addestramento che a dataset esterni, come citato in precedenza.

```

CLIENTS = ['miccai', 'bns']
CLIENTS_2 = [cl + '_2' for cl in CLIENTS]
#####
TOTAL_CLIENTS = len(CLIENTS)

DIR_DATA = 'data/imagesTrAug/'
DIR_GT = 'data/labelsTrBW/'

skin_dataset = dict()
skin_dataset['miccai'] = ['miccai_{:03d}'.format(i) for i in range(201, 227)]
skin_dataset['bns'] = ['bns_{:03d}'.format(i) for i in range(181, 204)]

split_dataset = dict()
STATIC_WEIGHT = [0,0]
order = 0

for client in skin_dataset:
    tmp = skin_dataset[client]
    x_ = [os.path.join(DIR_DATA, f + '.png') for f in tmp]
    y_ = [os.path.join(DIR_GT, f + '.png') for f in tmp]

    x_train, x_test, y_train, y_test = train_test_split(
        x_, y_, test_size=1 - TRAIN_RATIO, random_state=RS)

    split_dataset[client + '_train'] = Cancer(x_train, y_train, train=True, \
                                                IMAGE_SIZE=IMAGE_SIZE \
                                                , CROP_SIZE=CROP_SIZE)
    STATIC_WEIGHT[order] = len(x_train)
    order += 1

    split_dataset[client + '_test'] = Cancer(x_test, y_test, train=False, \
                                              IMAGE_SIZE=IMAGE_SIZE \
                                              , CROP_SIZE=CROP_SIZE)

print(client)

# Aggiungi le immagini rimanenti di Miccai al set di addestramento di Miccai
miccai_remaining = ['miccai_{:03d}'.format(i) for i in range(1, 201)] # Immagini non ancora aggiunte
miccai_train = split_dataset['miccai_train'].data # Percorsi delle immagini già nel set di addestramento di Miccai
miccai_remaining = [os.path.join(DIR_DATA, f + '.png') for f in miccai_remaining if f not in miccai_train]
split_dataset['miccai_train'].data.extend(miccai_remaining)

# Aggiungi le immagini rimanenti di Bns al set di addestramento di Bns
bns_remaining = ['bns_{:03d}'.format(i) for i in range(1, 181)] # Immagini non ancora aggiunte
bns_train = split_dataset['bns_train'].data # Percorsi delle immagini già nel set di addestramento di Bns
bns_remaining = [os.path.join(DIR_DATA, f + '.png') for f in bns_remaining if f not in bns_train]
split_dataset['bns_train'].data.extend(bns_remaining)

```

**Figura 3.8:** Sezione di codice per lo split delle immagini. La variabile TRAIN\_RATIO è impostata a 0.8. La classe Cancer gestisce il preprocessing delle immagini e delle ground truth.

### **3.4.2 Architettura U-Net**

L’architettura utilizzata è una variante della U-Net, un modello di rete neurale convoluzionale ampiamente utilizzato per la segmentazione di immagini biomediche. La U-Net è composta da due parti principali: il percorso di contrazione (downsampling) e il percorso di espansione (upsampling). Nel percorso di contrazione, l’immagine di input viene progressivamente ridotta di dimensione tramite operazioni di convoluzione e pooling. Questo percorso è progettato per estrarre le caratteristiche salienti dall’immagine in input [25]. Nel percorso di espansione, l’immagine ridotta viene riportata alla dimensione originale tramite operazioni di upsampling e concatenazione con le caratteristiche estratte nel percorso di contrazione. Questo percorso è progettato per consentire al modello di generare una mappa di segmentazione ad alta risoluzione basata sulle caratteristiche estratte nel percorso di contrazione. La U-Net utilizza anche connessioni residue tra i livelli del percorso di contrazione e di espansione per facilitare il flusso delle informazioni durante la fase di training. Inoltre, per ridurre il rischio di overfitting e migliorare la capacità del modello di generalizzare, vengono utilizzate tecniche di normalizzazione diversa rispetto alla batch normalization, come la group normalization, che ha dimostrato di funzionare meglio in questo contesto. L’output finale della U-Net è una mappa di segmentazione che assegna una classe a ciascun pixel dell’immagine di input, consentendo così la segmentazione accurata delle strutture di interesse. Questa architettura è particolarmente adatta per la segmentazione di immagini biomediche, dove la localizzazione precisa delle strutture è fondamentale per la diagnosi e il trattamento delle malattie.

### **3.4.3 Parametri di configurazione**

Durante l’implementazione dell’algoritmo di federated learning per la segmentazione delle immagini mediche, è stato cruciale definire accuratamente i parametri di configurazione per garantire prestazioni ottimali. Utilizzando le librerie principali come PyTorch, NumPy e Matplotlib, il sistema è stato configurato per massimizzare l’efficacia e la convergenza del modello. PyTorch è stato utilizzato per la definizione e l’addestramento del modello di segmentazione delle immagini mediche. Le funzionalità offerte da PyTorch hanno consentito di costruire reti neurali, ottimizzare i modelli e gestire i dati in modo efficiente. In particolare, sfruttando le funzionalità di PyTorch, è stato possibile definire il modello U-Net,

calcolare le perdite e aggiornare i pesi del modello durante l’addestramento [22]. NumPy è stata fondamentale per la manipolazione efficiente dei dati durante la fase di preprocessing e training. Le capacità di NumPy nel lavorare con array multidimensionali hanno permesso di gestire facilmente le immagini e le ground truth di segmentazione durante le operazioni di trasformazione e augmentazione dei dati. Matplotlib è stata utilizzata per la visualizzazione dei risultati dell’addestramento e delle prestazioni del modello [8]. Sfruttando le funzionalità di tracciamento di Matplotlib, sono stati creati grafici chiari e intuitivi per monitorare l’accuratezza e la perdita durante l’addestramento del modello su diverse epoche. Sono stati definiti diversi parametri chiave per ottimizzare il processo di addestramento del modello. Tra questi, il tasso di apprendimento (LR) è stato impostato a 0.001, determinando la velocità con cui i pesi del modello vengono aggiornati. Il weight decay (WD) è stato fissato a 0.0001, applicando una penalità alla norma dei pesi per prevenire l’overfitting. La soglia (TH) è stata utilizzata durante la generazione e il raffinamento delle pseudo-ground truth, assicurando la selezione di ground truth di alta qualità. Questi parametri sono stati attentamente calibrati per massimizzare la convergenza del modello e garantire risultati ottimali durante l’addestramento su dati distribuiti. L’importazione di moduli come os e copy ha consentito una gestione efficiente dei file e degli oggetti durante l’esecuzione dell’algoritmo di federated learning. Questi moduli sono stati utilizzati per la gestione dei percorsi dei file, la creazione di directory e la duplicazione di oggetti complessi come modelli di rete neurale. Inoltre, sono stati importati altri moduli come scikit-learn e Matplotlib per eseguire operazioni di split dei dati, preparazione dei dati e visualizzazione dei risultati. Questi moduli sono stati utilizzati per suddividere il dataset in training e testing set, valutare le prestazioni del modello e generare grafici per monitorare l’andamento dell’addestramento.

### 3.4.4 Ottimizzatore Adam

L’ottimizzatore Adam è uno degli algoritmi di ottimizzazione più utilizzati nell’addestramento dei modelli di deep learning. La sua popolarità deriva dalla sua efficacia nel convergere rapidamente verso i minimi locali della funzione di perdita, consentendo un addestramento più efficiente e stabile dei modelli. Adam combina le proprietà del momento, che tiene conto del gradiente passato per ag-

giornare i pesi del modello, e dell’adattamento del tasso di apprendimento, che regola dinamicamente il tasso di apprendimento per ciascun parametro del modello in base alla storia dei gradienti. Questa combinazione consente ad Adam di adattarsi in modo efficace a variazioni nei dati e nel paesaggio della loss function durante l’addestramento. Nell’implementazione dell’algoritmo di federated learning, l’ottimizzatore Adam è stato configurato con un tasso di apprendimento (LR) e un peso di decadimento (WD) appropriati. Questi parametri sono stati scelti con cura per garantire un buon equilibrio tra la velocità di convergenza del modello e la sua capacità di generalizzazione ai dati non visti. Un tasso di apprendimento troppo elevato potrebbe causare oscillazioni indesiderate durante l’addestramento, mentre un tasso troppo basso potrebbe rallentare eccessivamente la convergenza. Il peso di decadimento è stato utilizzato per applicare una penalità alla norma dei pesi del modello, contribuendo così alla regolarizzazione e alla prevenzione dell’overfitting [5].

### 3.4.5 Loss Function

Passando alla funzione di perdita, il coefficiente di Dice è stato scelto come misura di performance principale per valutare le prestazioni del modello durante l’addestramento. Il coefficiente di Dice è una misura di performance ampiamente utilizzata per valutare la similarità tra due insiemi di dati, in particolare nelle applicazioni di segmentazione delle immagini. Durante l’addestramento del modello di segmentazione delle immagini, l’obiettivo è massimizzare il coefficiente di Dice, il che significa massimizzare la similarità tra le maschere predette dal modello e le maschere di ground truth. Questo aiuta a garantire che il modello produca segmentazioni accurate e coerenti delle immagini mediche, essenziali per applicazioni diagnostica e terapeutica. La funzione di perdita basata sul coefficiente di Dice è stata implementata utilizzando la classe DiceCoeff [31].

### 3.4.6 Processo di data augmentation

Essendo le immagini utilizzate per il training di quantità molto bassa, come indicato nella Tabella 3.1, è stato deciso di impiegare un processo di data augmentation per ogni dataset utilizzato per il training (MICCAI, BNS e MoNuSeg). La data augmentation consente di aumentare il numero di dati effettuando delle tra-

```

# Applica la data augmentation alle immagini e alle etichette
for image_path, label_path in zip(image_files, label_files):
    # Carica l'immagine
    image = np.array(Image.open(image_path))
    # Carica l'etichetta di segmentazione
    label = np.array(Image.open(label_path))

    # Applica la data augmentation sia all'immagine che all'etichetta
    augmented = augmenter(image=image, segmentation_maps=SegmentationMapsOnImage(label, shape=image.shape))
    augmented_image = augmented[0]
    augmented_label = augmented[1].get_arr_int() # Converti in array numpy

    # Salva le immagini risultanti
    Image.fromarray(augmented_image).save(os.path.join(output_dir_images, f"monu_{total_images_created:03d}.png"))
    Image.fromarray(augmented_label).save(os.path.join(output_dir_labels, f"monu_{total_images_created:03d}.png"))

    total_images_created += 1

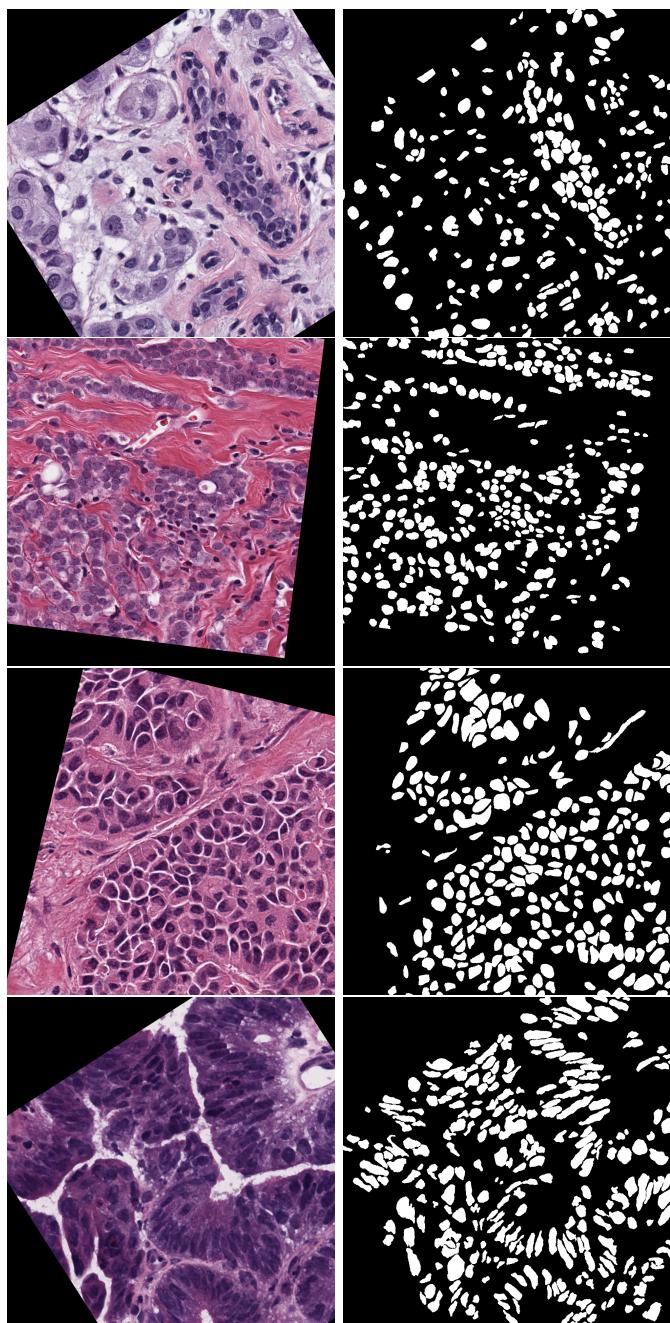
```

**Figura 3.9:** Sezione di codice per la data augmentation

sformazioni, per avere per ogni singolo dato, una moltitudine di varianti, in modo che il modello possa apprendere con più accuratezza e precisione [29] In questo caso, per ogni immagine, sono state effettuate delle trasformazioni con la libreria di Python imgaug. Tale libreria ha permesso di eseguire delle rotazioni, degli scaling, degli shear e delle traslazioni in range predefiniti. Allo stesso modo, le trasformazioni sono state applicate alle ground truth corrispondenti. Ciò ha permesso di ottenere esattamente 226 immagini ed ground truth MICCAI, 203 BNS e 154 MoNuSeg. Il codice per effettuare la data augmentation è mostrato in Figura 3.9. Quando si applicano trasformazioni di data augmentation a un'immagine e alle sue corrispondenti ground truth di segmentazione, è fondamentale che le trasformazioni siano applicate in modo sincrono sia all'immagine che alle ground truth. Se, ad esempio, si ruotasse un'immagine di 90 gradi, ci si deve assicurare che anche la ground truth di segmentazione sia ruotata di 90 gradi nello stesso modo, altrimenti non si perderebbe la corrispondenza immagine-ground truth, fondamentale per il training. La classe SegmentationMapsOnImage della libreria precedentemente citata gestisce questo sincronismo e facilita l'applicazione delle trasformazioni. Degli esempi di trasformazioni sono riportati in Figura 3.10.

**Tabella 3.1:** Dati delle immagini per dataset

Dataset	Originali	Augmented	Train	Validation	Test
MICCAI	26	226	220	6	6
BNS	18	203	198	5	5
MoNuSeg	51	154	143	11	11
TNBC	50	144	133	11	11
MoNuSAC	101	201	137	9	9



**Figura 3.10:** Esempi di trasformazioni applicate durante il processo di training.

### 3.5 Misure di performance dei modelli

Nel contesto della segmentazione delle immagini mediche, la valutazione della performance è comunemente effettuata con le misure, di seguito presentate e da

cui implementazione in Python è mostrata in Figura 3.11, applicate sui seguenti elementi ottenibili dall'applicazione di un metodo su immagini di test:

- **True Positive (TP)**: rappresenta il numero di pixel che appartengono ai nuclei cellulari (Foreground Tissue Units, FTU) che sono stati correttamente etichettati come tali.
- **False Positive (FP)**: rappresenta il numero di pixel di sfondo che sono stati erroneamente etichettati come pixel di nuclei cellulari.
- **False Negative (FN)**: rappresenta il numero di pixel FTU che sono stati erroneamente etichettati come pixel di sfondo.
- **True Negative (TN)**: rappresenta il numero di pixel di sfondo che sono stati correttamente etichettati come pixel di sfondo.

```
import numpy as np

def precision_score_(groundtruth_mask, pred_mask):
    intersect = np.sum(pred_mask*groundtruth_mask)
    total_pixel_pred = np.sum(pred_mask)
    precision = np.mean(intersect/total_pixel_pred)
    return round(precision, 3)

def recall_score_(groundtruth_mask, pred_mask):
    intersect = np.sum(pred_mask*groundtruth_mask)
    total_pixel_truth = np.sum(groundtruth_mask)
    recall = np.mean(intersect/total_pixel_truth)
    return round(recall, 3)

def accuracy(groundtruth_mask, pred_mask):
    intersect = np.sum(pred_mask*groundtruth_mask)
    union = np.sum(pred_mask) + np.sum(groundtruth_mask) - intersect
    xor = np.sum(groundtruth_mask==pred_mask)
    acc = np.mean(xor/(union + xor - intersect))
    return round(acc, 3)

def dice_coef(groundtruth_mask, pred_mask):
    intersect = np.sum(pred_mask*groundtruth_mask)
    total_sum = np.sum(pred_mask) + np.sum(groundtruth_mask)
    dice = np.mean(2*intersect/total_sum)
    return round(dice, 3) #round up to 3 decimal places

def iou(groundtruth_mask, pred_mask):
    intersect = np.sum(pred_mask*groundtruth_mask)
    union = np.sum(pred_mask) + np.sum(groundtruth_mask) - intersect
    iou = np.mean(intersect/union)
    return round(iou, 3)
```

**Figura 3.11:** Implementazione in Python delle misure di performance<sup>3</sup>.

---

<sup>3</sup>Understanding Evaluation Metrics in Medical Image Segmentation

## **Accuratezza**

L'accuracy, o accuratezza, è una misura fondamentale delle prestazioni del modello di segmentazione delle immagini mediche. Essa rappresenta la proporzione di predizioni corrette rispetto al totale delle predizioni effettuate. L'accuracy viene calcolata come il rapporto tra il numero di veri positivi e veri negativi (predizioni corrette) e il numero totale di campioni. Questo parametro è utile per ottenere una visione generale delle capacità del modello, ma potrebbe non essere sufficiente da solo, soprattutto in contesti dove le classi sono sbilanciate. Ad esempio, in un dataset medico dove la presenza della patologia è rara, un modello che predice sempre l'assenza della patologia potrebbe comunque avere un'alta accuracy, pur non essendo efficace.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

## **Precisione**

La precision, o precisione, è una misura di performance che valuta la qualità delle predizioni positive del modello. Viene calcolata come il rapporto tra il numero di veri positivi e la somma dei veri positivi e dei falsi positivi. Una precisione elevata indica che il modello ha un basso tasso di falsi positivi, ossia è molto affidabile quando predice la presenza di una certa classe. In ambito medico, una precisione alta significa che la maggior parte delle lesioni identificate dal modello sono effettivamente corrette, riducendo il rischio di falsi allarmi e ulteriori test diagnostici inutili.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

## **Recall**

Il recall, o sensibilità, è una misura di performance che valuta la capacità del modello di identificare correttamente le istanze positive. Viene calcolata come il rapporto tra il numero di veri positivi e la somma dei veri positivi e dei falsi negativi. Un recall elevato indica che il modello riesce a catturare la maggior parte delle istanze positive, riducendo il numero di falsi negativi. In contesti clinici, un alto recall è cruciale, poiché assicura che pochi casi di patologia vengano persi, aumentando l'efficacia del modello nel rilevare malattie o condizioni cliniche.

critiche.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

### Dice Coefficient

Il Dice Coefficient è una misura di similarità utilizzata per valutare le prestazioni dei modelli di segmentazione. È particolarmente utile in contesti dove si vuole misurare l'accordo tra due set di dati binari. Viene calcolato come il doppio dell'intersezione tra le predizioni e le ground truth vere, diviso per la somma delle predizioni e delle ground truth. Il valore del Dice Coefficient varia tra 0 e 1, dove 1 indica una perfetta sovrapposizione tra le predizioni e le ground truth. Questo coefficiente è particolarmente utile per valutare la segmentazione di strutture mediche, poiché penalizza sia i falsi positivi che i falsi negativi.

$$\text{Dice Coefficient} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}$$

### Test Loss

La test loss rappresenta una misura del costo associato alle predizioni del modello su un set di dati di test. Durante l'addestramento del modello, la funzione di perdita viene minimizzata per migliorare le prestazioni del modello. La test loss è calcolata confrontando le predizioni del modello con le ground truth reali nel set di test e applicando una funzione di perdita predefinita, come la Cross-Entropy Loss o la Dice Loss. Una bassa test loss indica che il modello ha una buona capacità di generalizzazione e che è in grado di effettuare predizioni accurate su dati non visti [3]. In questo lavoro, la Test Loss utilizzata è stata la Dice Loss, la cui definizione è espressa come:

$$\text{Test Loss} = 1 - \text{Dice Coefficient}$$

### Jaccard Index

Il Jaccard Index, noto anche come Intersection over Union (IoU), è un'altra misura di similarità utilizzata per valutare la qualità della segmentazione. Viene calcolato come il rapporto tra l'intersezione delle predizioni e delle ground truth e l'unione delle predizioni e delle ground truth. Il valore del Jaccard Index varia tra 0 e 1, con 1 che rappresenta una perfetta sovrapposizione. Questa misura

di performance è ampiamente utilizzata in molte applicazioni di segmentazione, compresa l'analisi delle immagini mediche, poiché fornisce una misura chiara e intuitiva di quanto le predizioni del modello coincidano con le ground truth reali [27].

$$\text{Jaccard Index} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}$$



# Capitolo 4

## Risultati sperimentali

In questo capitolo verranno elencati ed analizzati tutti i risultati prodotti dai modelli, messi alla prova sia su test set interni in Sezione 4.1, sia su test set esterni in Sezione 4.2. Inoltre, verrà effettuato un primo affinamento di apprendimento incrementale e ne verranno valutate le prestazioni in Sezione 4.3 per test set interni e in Sezione 4.4 per test set esterni. In seguito, verrà eseguito un ulteriore affinamento cross-dataset in Sezione 4.5. Infine, verranno comparati tutti i risultati ottenuti in Sezione 4.6, dei quali si discuterà in dettaglio in Sezione 4.7.

### 4.1 Risultati in-dataset

Gli esperimenti riguardano tutte le tecniche di apprendimento precedentemente citate: FedAvg, FedMix, FedST, FedRGD e Local Learning. Inizialmente, dopo aver addestrato i modelli, sono stati testati sui test set degli stessi dataset utilizzati per il training.

I risultati di FedAvg sono mostrati in Tabella 4.1.

FedAvg	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI	0.936	<b>0.809</b>	<b>0.848</b>	<b>0.179</b>	<b>0.821</b>	<b>0.701</b>
BNS	<b>0.969</b>	0.794	0.810	0.204	0.797	0.665
Fed on MICCAI	0.930	0.772	<b>0.874</b>	<b>0.190</b>	<b>0.811</b>	<b>0.687</b>
Fed on BNS	<b>0.968</b>	<b>0.852</b>	0.753	0.210	0.792	0.659

**Tabella 4.1:** Risultati delle diverse prestazioni di FedAvg. Sono evidenziati in grassetto i due migliori risultati per misura di performance.

I risultati di FedMix sono mostrati in Tabella 4.2.

FedMix	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI	0.899	<b>0.763</b>	0.724	<b>0.293</b>	<b>0.719</b>	<b>0.572</b>
BNS	<b>0.928</b>	0.465	<b>0.800</b>	0.672	0.484	0.341
Fed on MICCAI	0.902	<b>0.788</b>	0.695	<b>0.358</b>	<b>0.718</b>	<b>0.570</b>
Fed on BNS	<b>0.921</b>	0.420	<b>0.833</b>	0.677	0.459	0.322

**Tabella 4.2:** Risultati delle diverse prestazioni di FedMix. Sono evidenziati in grassetto i due migliori risultati per misura di performance.

I risultati di FedST sono mostrati in Tabella 4.3.

FedST	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI	<b>0.191</b>	<b>0.191</b>	<b>1</b>	<b>0.732</b>	<b>0.314</b>	<b>0.191</b>
BNS	0.074	0.074	<b>1</b>	0.888	0.127	0.074
Fed on MICCAI	<b>0.191</b>	<b>0.191</b>	<b>1</b>	<b>0.732</b>	<b>0.314</b>	<b>0.191</b>
Fed on BNS	0.074	0.074	<b>1</b>	0.888	0.127	0.074

**Tabella 4.3:** Risultati delle diverse prestazioni di FedST. Sono evidenziati in grassetto i due migliori risultati per misura di performance.

I risultati di FedRGD sono mostrati in Tabella 4.4.

FedRGD	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI	<b>0.191</b>	<b>0.191</b>	<b>1</b>	<b>0.732</b>	<b>0.314</b>	<b>0.191</b>
BNS	0.074	0.074	<b>1</b>	0.888	0.127	0.074
Fed on MICCAI	<b>0.191</b>	<b>0.191</b>	<b>1</b>	<b>0.732</b>	<b>0.314</b>	<b>0.191</b>
Fed on BNS	0.074	0.074	<b>1</b>	0.888	0.127	0.074

**Tabella 4.4:** Risultati delle diverse prestazioni di FedRGD. Sono evidenziati in grassetto i due migliori risultati per misura di performance.

I risultati di Local Learning sono mostrati in Tabella 4.5.

LL	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI	0.924	0.774	<b>0.853</b>	<b>0.205</b>	<b>0.796</b>	<b>0.669</b>
BNS	<b>0.965</b>	<b>0.768</b>	0.764	0.255	0.746	0.603

**Tabella 4.5:** Risultati delle diverse prestazioni di LL. Sono evidenziati in grassetto il miglior risultato per misura di performance.

## 4.2 Risultati su dataset out-of-distribution

Valutare i modelli su dataset esterni (out-of-distribution, OOD) è cruciale per comprendere appieno le loro prestazioni al di fuori del contesto in cui sono stati addestrati. I test in-dataset forniscono una valutazione delle capacità del modello su dati simili a quelli usati per l'addestramento, ma non garantiscono che il modello sia in grado di generalizzare su nuovi dati provenienti da altre fonti o contesti. I dataset esterni consentono di esplorare la robustezza e l'efficacia del modello in scenari più diversificati e realistici, aiutando a identificare eventuali limitazioni o aree di miglioramento.

Questi esperimenti mirano a fornire una panoramica più completa delle capacità predittive dei modelli, confrontando le prestazioni su dati in-dataset con quelle su dati esterni. Questa comparazione aiuta a valutare la capacità di generalizzazione dei modelli e a fornire indicazioni importanti per il miglioramento delle prestazioni in contesti reali. Di seguito vi è l'elenco dei risultati ottenuti utilizzando come test set dei dataset esterni.

I risultati di FedAvg sono mostrati in Tabella 4.6; quelli di FedMix sono mostrati in Tabella 4.7; quelli di FedST sono mostrati in Tabella 4.8; quelli di risultati di FedRGD sono mostrati in Tabella 4.9 e infine quelli di Local Learning sono mostrati in Tabella 4.10.

FedAvg	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on MoNuSeg	<b>0.899</b>	<b>0.732</b>	0.838	<b>0.221</b>	<b>0.779</b>	<b>0.64</b>
BNS on MoNuSeg	0.869	0.638	<b>0.893</b>	0.259	0.742	0.592
Fed on MoNuSeg	0.892	0.7	0.873	0.227	0.774	0.632
MICCAI on TNBC	0.929	0.777	0.617	0.336	0.664	0.515
BNS on TNBC	<b>0.934</b>	0.753	<b>0.685</b>	<b>0.305</b>	<b>0.695</b>	<b>0.55</b>
Fed on TNBC	0.933	<b>0.792</b>	0.624	0.328	0.673	0.524
MICCAI on MoNuSAC	<b>0.788</b>	<b>0.72</b>	0.491	0.461	0.539	0.394
BNS on MoNuSAC	0.785	0.685	<b>0.54</b>	<b>0.446</b>	<b>0.554</b>	<b>0.409</b>
Fed on MoNuSAC	0.785	0.713	0.497	0.465	0.535	0.392

**Tabella 4.6:** Risultati delle prestazioni dei modelli FedAvg su dataset di test esterni. Sono evidenziati in grassetto i migliori risultati per misura di performance all'interno di ogni dataset.

FedMix	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on MoNuSeg	0.889	0.747	0.747	0.27	0.74	0.588
BNS on MoNuSeg	0.894	0.816	0.672	0.329	0.73	0.577
Fed on MoNuSeg	<b>0.896</b>	<b>0.785</b>	<b>0.725</b>	<b>0.318</b>	<b>0.747</b>	<b>0.598</b>
MICCAI on TNBC	0.895	0.56	<b>0.73</b>	<b>0.46</b>	<b>0.557</b>	<b>0.407</b>
BNS on TNBC	<b>0.902</b>	<b>0.618</b>	0.648	0.551	0.554	0.4
Fed on TNBC	0.898	0.582	0.694	0.544	<b>0.557</b>	0.404
MICCAI on MoNuSAC	0.766	0.668	<b>0.473</b>	0.492	<b>0.513</b>	<b>0.36</b>
BNS on MoNuSAC	0.763	<b>0.704</b>	0.394	<b>0.542</b>	0.467	0.318
Fed on MoNuSAC	<b>0.767</b>	0.691	0.434	0.52	0.494	0.342

**Tabella 4.7:** Risultati delle prestazioni dei modelli FedMix su dataset di test esterni. Sono evidenziati in grassetto i migliori risultati per misura di performance all'interno di ogni dataset.

FedST	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on MoNuSeg	0.216	0.216	1	0.702	0.352	0.216
BNS on MoNuSeg	0.216	0.216	1	0.702	0.352	0.216
Fed on MoNuSeg	0.216	0.216	1	0.702	0.352	0.216
MICCAI on TNBC	0.125	0.125	1	0.816	0.211	0.125
BNS on TNBC	0.125	0.125	1	0.816	0.211	0.125
Fed on TNBC	0.125	0.125	1	0.816	0.211	0.125
MICCAI on MoNuSAC	0.302	0.302	1	0.648	0.441	0.302
BNS on MoNuSAC	0.302	0.302	1	0.648	0.441	0.302
Fed on MoNuSAC	0.302	0.302	1	0.648	0.441	0.302

**Tabella 4.8:** Risultati delle prestazioni dei modelli FedST su dataset di test esterni.

### 4.3 Risultati apprendimento incrementale in-dataset

Dopo aver addestrato i modelli con due client (MICCAI e BNS), si è deciso di effettuare un ulteriore addestramento con le due metodologie che hanno prodotto i risultati complessivamente migliori: FedAvg e FedMix. Quello che è stato effettuato è un approccio di apprendimento incrementale, in cui si estende il numero di client per poter verificare eventuali miglioramenti nelle prestazioni. Nello specifico, il dataset che è stato aggiunto ai client di addestramento è MoNuSeg.

I risultati di FedAvg sono mostrati in Tabella 4.11, mentre i risultati di FedMix sono mostrati in Tabella 4.12.

FedRGD	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on MoNuSeg	0.216	0.216	1	0.702	0.352	0.216
BNS on MoNuSeg	0.216	0.216	1	0.702	0.352	0.216
Fed on MoNuSeg	0.216	0.216	1	0.702	0.352	0.216
MICCAI on TNBC	0.125	0.125	1	0.816	0.211	0.125
BNS on TNBC	0.125	0.125	1	0.816	0.211	0.125
Fed on TNBC	0.125	0.125	1	0.816	0.211	0.125
MICCAI on MoNuSAC	0.302	0.302	1	0.648	0.441	0.302
BNS on MoNuSAC	0.302	0.302	1	0.648	0.441	0.302
Fed on MoNuSAC	0.302	0.302	1	0.648	0.441	0.302

**Tabella 4.9:** Risultati delle prestazioni dei modelli FedRGD su dataset di test esterni.

LL	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on MoNuSeg	<b>0.891</b>	<b>0.73</b>	0.794	<b>0.246</b>	<b>0.755</b>	<b>0.608</b>
BNS on MoNuSeg	0.852	0.604	<b>0.896</b>	0.286	0.716	0.561
MICCAI on TNBC	0.908	0.605	0.604	0.447	0.553	0.402
BNS on TNBC	<b>0.932</b>	<b>0.771</b>	<b>0.648</b>	<b>0.322</b>	<b>0.679</b>	<b>0.532</b>
MICCAI on MoNuSAC	<b>0.785</b>	<b>0.713</b>	0.513	<b>0.446</b>	<b>0.554</b>	<b>0.403</b>
BNS on MoNuSAC	0.757	0.627	<b>0.562</b>	0.456	0.545	0.395

**Tabella 4.10:** Risultati delle prestazioni dei modelli LL su dataset di test esterni. Sono evidenziati in grassetto i migliori risultati per misura di performance all'interno di ogni dataset.

## 4.4 Risultati apprendimento incrementale su dataset OOD

Di seguito verranno elencati i risultati sui test set esterni. Nello specifico, i risultati di FedAvg sono mostrati in Tabella 4.13, mentre i risultati di FedMix sono mostrati in Tabella 4.14.

FedAvg	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI	0.913	0.713	<b>0.903</b>	0.221	0.779	0.649
BNS	<b>0.939</b>	0.447	<b>0.891</b>	0.467	0.552	0.406
MoNuSeg	0.846	0.638	0.771	0.31	0.69	0.556
Fed on MICCAI	0.934	<b>0.81</b>	0.831	<b>0.186</b>	<b>0.815</b>	<b>0.691</b>
Fed on BNS	<b>0.966</b>	<b>0.868</b>	0.736	<b>0.214</b>	<b>0.789</b>	<b>0.657</b>
Fed on MoNuSeg	0.827	0.593	0.812	0.323	0.677	0.538

**Tabella 4.11:** Risultati delle diverse prestazioni di FedAvg. Sono evidenziati in grassetto i due migliori risultati per misura di performance.

FedMix	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI	<b>0.806</b>	<b>0.502</b>	0.941	<b>0.37</b>	<b>0.631</b>	<b>0.477</b>
BNS	0.739	0.191	<b>0.949</b>	0.733	0.268	0.176
MoNuSeg	0.704	0.391	0.814	0.487	0.513	0.354
Fed on MICCAI	<b>0.808</b>	<b>0.504</b>	0.940	<b>0.368</b>	<b>0.633</b>	<b>0.479</b>
Fed on BNS	0.735	0.189	<b>0.951</b>	0.734	0.267	0.175
Fed on MoNuSeg	0.71	0.396	0.808	0.484	0.517	0.358

**Tabella 4.12:** Risultati delle diverse prestazioni di FedMix. Sono evidenziati in grassetto i due migliori risultati per misura di performance e dataset.

FedAvg	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on TNBC	<b>0.929</b>	<b>0.729</b>	<b>0.691</b>	<b>0.315</b>	<b>0.687</b>	<b>0.538</b>
BNS on TNBC	0.897	0.53	<b>0.841</b>	0.384	0.617	0.468
MoNuSeg on TNBC	0.913	0.72	0.505	0.439	0.562	0.401
Fed on TNBC	<b>0.93</b>	<b>0.794</b>	0.596	<b>0.341</b>	<b>0.659</b>	<b>0.505</b>
MICCAI on MoNuSAC	<b>0.788</b>	0.667	<b>0.583</b>	<b>0.42</b>	<b>0.58</b>	<b>0.433</b>
BNS on MoNuSAC	<b>0.785</b>	0.664	<b>0.556</b>	<b>0.441</b>	<b>0.559</b>	<b>0.413</b>
MoNuSeg on MoNuSAC	0.772	<b>0.706</b>	0.417	0.518	0.482	0.338
Fed on MoNuSAC	0.783	<b>0.696</b>	0.481	0.476	0.524	0.38

**Tabella 4.13:** Risultati delle diverse prestazioni di FedAvg su dati esterni. Sono evidenziati in grassetto i due migliori risultati per misura di performance e dataset esterno.

FedMix	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on TNBC	0.77	0.328	<b>0.921</b>	0.561	0.44	0.306
BNS on TNBC	<b>0.773</b>	<b>0.33</b>	0.919	<b>0.56</b>	<b>0.441</b>	<b>0.307</b>
MoNuSeg on TNBC	0.767	0.325	<b>0.924</b>	0.563	0.438	0.304
Fed on TNBC	<b>0.772</b>	<b>0.33</b>	0.92	<b>0.559</b>	<b>0.442</b>	<b>0.308</b>
MICCAI on MoNuSAC	<b>0.75</b>	<b>0.564</b>	0.724	<b>0.408</b>	<b>0.592</b>	<b>0.447</b>
BNS on MoNuSAC	0.748	0.561	<b>0.725</b>	0.41	0.591	0.445
MoNuSeg on MoNuSAC	0.749	<b>0.564</b>	<b>0.728</b>	<b>0.407</b>	<b>0.593</b>	<b>0.448</b>
Fed on MoNuSAC	<b>0.75</b>	<b>0.566</b>	0.722	<b>0.408</b>	<b>0.592</b>	<b>0.447</b>

**Tabella 4.14:** Risultati delle diverse prestazioni di FedMix su dati esterni. Sono evidenziati in grassetto i due migliori risultati per misura di performance e dataset esterno.

## 4.5 Sperimentazione cross-dataset

Un’ulteriore valutazione è stata condotta con un setup cross-dataset per tutti i modelli finora addestrati. La sperimentazione cross-dataset è essenziale per valutare la capacità dei modelli di generalizzare su dati provenienti da diverse fonti o contesti. Questo tipo di esperimento permette di testare se le prestazioni osservate sui dataset di addestramento si traducono in un buon adattamento su nuovi dataset, contribuendo così a determinare la robustezza e l'affidabilità del modello in scenari reali.

I risultati di FedAvg (2 Clients) sono mostrati in Tabella 4.15.

FedAvg (2 Clients)	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on BNS	<b>0.964</b>	<b>0.784</b>	0.76	0.242	0.759	0.618
BNS on MICCAI	0.911	0.695	<b>0.9</b>	<b>0.228</b>	<b>0.773</b>	<b>0.639</b>

**Tabella 4.15:** Risultati delle diverse prestazioni di FedAvg (2 Clients) per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance.

I risultati di FedMix (2 Clients) sono mostrati in Tabella 4.16.

FedMix (2 Clients)	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on BNS	<b>0.912</b>	0.392	<b>0.851</b>	0.588	0.435	0.305
BNS on MICCAI	0.9	<b>0.819</b>	0.642	<b>0.37</b>	<b>0.7</b>	<b>0.549</b>

**Tabella 4.16:** Risultati delle diverse prestazioni di FedMix (2 Clients) per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance.

I risultati di FedST sono mostrati in Tabella 4.17.

FedST	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on BNS	0.074	0.074	<b>1</b>	0.888	0.127	0.074
BNS on MICCAI	<b>0.191</b>	<b>0.191</b>	<b>1</b>	<b>0.732</b>	<b>0.314</b>	<b>0.191</b>

**Tabella 4.17:** Risultati delle diverse configurazioni di FedST per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance.

I risultati di FedRGD sono mostrati in Tabella 4.18.

FedRGD	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on BNS	0.074	0.074	<b>1</b>	0.888	0.127	0.074
BNS on MICCAI	<b>0.191</b>	<b>0.191</b>	<b>1</b>	<b>0.732</b>	<b>0.314</b>	<b>0.191</b>

**Tabella 4.18:** Risultati delle diverse configurazioni di FedRGD per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance.

I risultati di Local Learning sono mostrati in Tabella 4.19.

LL	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on BNS	<b>0.926</b>	0.447	0.615	0.57	0.431	0.294
BNS on MICCAI	0.896	<b>0.679</b>	<b>0.867</b>	<b>0.258</b>	<b>0.743</b>	<b>0.602</b>

**Tabella 4.19:** Risultati delle diverse configurazioni di LL per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance.

I risultati di FedAvg (3 Clients) sono mostrati in Tabella 4.20.

FedAvg (3 Clients)	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on BNS	<b>0.964</b>	<b>0.776</b>	0.798	<b>0.229</b>	<b>0.774</b>	<b>0.638</b>
BNS on MICCAI	0.915	0.713	0.891	<b>0.219</b>	<b>0.782</b>	<b>0.65</b>
MICCAI on MoNuSeg	0.872	0.641	<b>0.906</b>	0.254	0.747	0.598
BNS on MoNuSeg	0.859	0.624	<b>0.922</b>	0.266	0.734	0.587
MoNuSeg on MICCAI	0.917	<b>0.834</b>	0.683	0.256	0.746	0.601
MoNuSeg on BNS	<b>0.956</b>	0.775	0.687	0.31	0.694	0.542

**Tabella 4.20:** Risultati delle diverse prestazioni di FedAvg (3 Clients) per la sperimentazione cross-dataset. Sono evidenziati in grassetto i due risultati migliori per misura di performance.

I risultati di FedMix (3 Clients) sono mostrati in Tabella 4.21.

FedMix (3 Clients)	Acc	Prec	Rec	T. Loss	Dice	IoU
MICCAI on BNS	0.732	0.188	<b>0.952</b>	0.735	0.265	0.174
BNS on MICCAI	<b>0.803</b>	<b>0.498</b>	0.938	<b>0.375</b>	<b>0.626</b>	<b>0.473</b>
MICCAI on MoNuSeg	0.709	0.395	0.811	0.484	0.516	0.357
BNS on MoNuSeg	0.707	0.393	0.805	0.488	0.514	0.354
MoNuSeg on MICCAI	<b>0.803</b>	<b>0.498</b>	0.943	<b>0.373</b>	<b>0.628</b>	<b>0.474</b>
MoNuSeg on BNS	0.727	0.186	<b>0.953</b>	0.737	0.263	0.173

**Tabella 4.21:** Risultati delle diverse prestazioni di FedMix (3 Clients) per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance.

## 4.6 Comparazione conclusiva

In questa sezione vengono confrontati i risultati ottenuti da tutte le tecniche di apprendimento considerate: FedAvg, FedMix, FedST, FedRGD e Local Learning. I risultati sono presentati sia sui test set interni (gli stessi utilizzati per l’addestramento) sia su test set esterni (dataset non utilizzati per l’addestramento).

**Tabella 4.22:** *Migliori risultati per ogni metodo sul dataset MICCAI*

Metodo	Acc	Prec	Rec	T. Loss	Dice	IoU
FedAvg (2)	<b>0.936</b>	0.809	0.848	<b>0.179</b>	<b>0.821</b>	<b>0.701</b>
FedMix (2)	0.899	0.763	0.724	0.293	0.719	0.572
FedAvg (3)	0.917	<b>0.834</b>	0.683	0.256	0.746	0.601
FedMix (3)	0.803	0.498	0.943	0.373	0.628	0.474
FedST	0.191	0.191	<b>1</b>	0.732	0.314	0.191
FedRGD	0.191	0.191	<b>1</b>	0.732	0.314	0.191
Local Learning	0.924	0.774	0.853	0.205	0.796	0.669

**Tabella 4.23:** *Migliori risultati per ogni metodo sul dataset BNS*

Metodo	Acc	Prec	Rec	T. Loss	Dice	IoU
FedAvg (2)	0.899	0.732	0.838	<b>0.221</b>	<b>0.779</b>	<b>0.64</b>
FedMix (2)	0.928	0.465	0.8	0.672	0.484	0.341
FedAvg (3)	0.964	<b>0.776</b>	0.798	0.229	0.774	0.638
FedMix (3)	0.732	0.188	0.952	0.735	0.265	0.174
FedST	0.074	0.074	<b>1</b>	0.888	0.127	0.074
FedRGD	0.074	0.074	<b>1</b>	0.888	0.127	0.074
Local Learning	<b>0.965</b>	0.768	0.764	0.255	0.746	0.603

**Tabella 4.24:** *Migliori risultati per ogni metodo sul dataset MoNuSeg*

Metodo	Acc	Prec	Rec	T. Loss	Dice	IoU
FedAvg (2)	<b>0.899</b>	0.732	0.838	<b>0.221</b>	<b>0.779</b>	<b>0.64</b>
FedMix (2)	0.896	<b>0.785</b>	0.725	0.318	0.747	0.598
FedAvg (3)	0.872	0.641	0.906	0.254	0.747	0.598
FedMix (3)	0.709	0.395	0.811	0.484	0.516	0.357
FedST	0.216	0.216	<b>1</b>	0.702	0.352	0.216
FedRGD	0.216	0.216	<b>1</b>	0.702	0.352	0.216
Local Learning	0.891	0.73	0.794	0.246	0.755	0.608

**Tabella 4.25:** *Migliori risultati per ogni metodo sul dataset TNBC*

Metodo	Acc	Prec	Rec	T. Loss	Dice	IoU
FedAvg (2)	<b>0.934</b>	0.753	0.685	<b>0.305</b>	<b>0.695</b>	<b>0.55</b>
FedMix (2)	0.902	0.618	0.648	0.551	0.554	0.4
FedAvg (3)	0.93	<b>0.794</b>	0.596	0.341	0.659	0.505
FedMix (3)	0.773	0.33	0.919	0.56	0.441	0.307
FedST	0.125	0.125	<b>1</b>	0.816	0.211	0.125
FedRGD	0.125	0.125	<b>1</b>	0.816	0.211	0.125
Local Learning	0.932	0.771	0.648	0.322	0.679	0.532

**Tabella 4.26:** *Migliori risultati per ogni metodo sul dataset MoNuSAC*

Metodo	Acc	Prec	Rec	T. Loss	Dice	IoU
FedAvg (2)	<b>0.788</b>	<b>0.72</b>	0.491	0.461	0.539	0.394
FedMix (2)	0.767	0.691	0.434	0.52	0.494	0.342
FedAvg (3)	0.785	0.664	0.556	0.441	0.559	0.413
FedMix (3)	0.75	0.566	0.722	<b>0.408</b>	<b>0.592</b>	<b>0.447</b>
FedST	0.302	0.302	<b>1</b>	0.648	0.441	0.302
FedRGD	0.302	0.302	<b>1</b>	0.648	0.441	0.302
Local Learning	0.785	0.713	0.513	0.446	0.554	0.403

## 4.7 Discussione

Nel contesto della segmentazione di nuclei cellulari, e di questo lavoro di tesi, l’obiettivo principale è sviluppare modelli di intelligenza artificiale in grado di identificare con precisione i confini dei nuclei nelle immagini istopatologiche. Questo è cruciale per una serie di applicazioni mediche, tra cui la diagnosi precoce del cancro e la valutazione della risposta al trattamento. Dai risultati ottenuti, possiamo osservare che quasi tutti i modelli si sono rivelati molto efficaci e con grandi potenzialità. Le performance mostrano valori maggiori dello 0.8 di accuratezza e ottimi valori anche per le altre misure di performance, rappresentando dei risultati importanti in contesti in cui un client non ha conoscenza dei dataset degli altri client.

FedAvg emerge come una tecnica di apprendimento federato promettente. I risultati mostrano una buona capacità di adattamento sui dati dei client, come indicato dalle alte percentuali di accuratezza, precisione e recall sui dataset

MICCAI e BNS. Tuttavia, l’efficacia di FedAvg su dataset esterni, come MoNu-Seg e TNBC, presenta variazioni significative, suggerendo possibili problemi di generalizzazione e overfitting. Con la sperimentazione di apprendimento incrementale, le prestazioni ottenute sui dataset esterni sono migliorate. Inoltre, le performance cross-dataset di FedAvg mostrano una certa robustezza, con risultati incoraggianti nell’applicazione su dataset diversi da quelli di addestramento.

D’altra parte, FedMix mostra risultati più eterogenei, con performance variabili su diversi dataset e una gestione meno efficace dell’eterogeneità dei dati rispetto a FedAvg. La sperimentazione di apprendimento incrementale ha fornito risultati ben più promettenti sul dataset MICCAI. Le percentuali di accuratezza, precisione e recall dipendono notevolmente dal dataset di riferimento, suggerendo la necessità di ulteriori ottimizzazioni per migliorare la capacità di generalizzazione del modello.

Come termine di confronto, il LL ha mostrato risultati piuttosto elevati, con una buona adattabilità ai dataset dei client ma una maggiore variabilità nelle performance su dataset esterni e cross-dataset, condizionate dal suo setup locale.

FedST e FedRGD, d’altro canto, non sono riusciti ad adattarsi efficacemente ad alcun dataset e hanno prodotto dei risultati scadenti, ad eccezione di una recall pari a 1 che, tuttavia, potrebbe dipendere dall’assenza di regioni localizzate dai modelli nelle immagini, quindi dal fatto che il modello non abbia potuto produrre falsi negativi poiché inadatto a compiere il task di segmentazione.

Infine, la generazione di modelli federati è stata senza ombra di dubbio un’ottima scelta. Essi sono stati in grado di generalizzare in modo opportuno e accurato tutti i modelli. Spesso, hanno persino prodotto i migliori risultati sia per dataset interni che esterni.

Le possibili cause di ottimi o pessimi risultati possono essere attribuite alla diversità e alla complessità dei dati, alla capacità dei modelli di adattarsi a diverse distribuzioni di dati e alla presenza di rumore e variazioni nelle immagini istopatologiche. Inoltre, la scelta degli iperparametri e delle tecniche di ottimizzazione potrebbe influenzare significativamente le performance dei modelli e merita un’ulteriore analisi approfondita.

# Capitolo 5

## Conclusioni e sviluppi futuri

Questa tesi ha esplorato diversi approcci di apprendimento federato per la segmentazione di immagini mediche, confrontando i metodi FedAvg, FedMix, FedST, FedRGD e LL su differenti dataset, con setup sia in-dataset che OOD, e analizzando le loro prestazioni in termini di accuratezza, precisione, recall, Dice loss, coefficiente di Dice e indice di Jaccard.

FedAvg ha mostrato risultati generalmente robusti su più dataset. In particolare, il metodo FedAvg ha raggiunto un'accuratezza massima di 0.969 con il dataset BNS, e una precisione e recall elevate rispettivamente con valori di 0.852 e 0.9 nel test su MICCAI e BNS. Questo metodo ha dimostrato un equilibrio solido tra accuratezza e recall, specialmente nei test incrociati, con recall pari a 0.922 su MoNuSeg. Tuttavia, i risultati su MoNuSAC sono stati inferiori rispetto ad altri dataset, evidenziando una potenziale limitazione nella generalizzazione del modello. Su dati esterni, FedAvg ha mostrato una buona robustezza: su MoNuSeg, l'accuratezza variava tra 0.86 e 0.89, mentre su TNBC, raggiungeva lo 0.93. Tuttavia, su MoNuSAC, le prestazioni erano inferiori, con un'accuratezza intorno allo 0.78. Nel contesto cross-dataset, FedAvg ha mostrato prestazioni solide, con un'accuratezza del 0.96 del modello MICCAI su dati BNS e dello 0.91 del modello BNS su dati MICCAI.

FedMix ha prodotto risultati eterogenei. Sebbene accuratezza e recall fossero significativamente alti in alcuni casi, come dimostrato dal valore di recall di 0.953 per FedMix su BNS, la precisione è risultata spesso inferiore, con un valore di 0.186 sullo stesso dataset. Questo indica che, sebbene il modello riesca a identificare correttamente molte istanze positive, ha difficoltà nel ridurre i falsi positivi,

quindi il modello tende a identificare delle regioni nucleari che in realtà non sono presenti. Inoltre, il metodo ha mostrato prestazioni subottimali nei test incrociati con TNBC e MoNuSAC, suggerendo che la capacità di generalizzazione potrebbe essere limitata.

FedST e FedRGD hanno entrambi presentato prestazioni significativamente inferiori rispetto agli altri metodi. I valori costantemente bassi di accuratezza, precisione e recall indicano che questi approcci non sono efficaci per i compiti di segmentazione considerati in questa tesi. Ad esempio, FedRGD ha ottenuto una precisione di soli 0.191 su MICCAI e una recall di 1, il che indica un modello altamente sbilanciato e non pratico per applicazioni reali.

Considerando il suo setup locale, LL ha prodotto buone prestazioni in-dataset, con un'accuratezza dello 0.92 su MICCAI e dello 0.96 su BNS. Tuttavia, sui dati esterni, le performance di LL sono state variabili: su MoNuSeg, LL ha ottenuto un'accuratezza dello 0.89 con il modello addestrato su MICCAI e dello 0.85 con quello su BNS. Su TNBC, le performance erano migliori con un'accuratezza dello 0.9 per MICCAI e dello 0.93 per BNS. Su MoNuSAC, l'accuratezza era dello 0.78 per MICCAI e dello 0.75 per BNS. Nel contesto cross-dataset, LL ha mostrato le seguenti prestazioni: il modello MICCAI su dati BNS ha ottenuto un'accuratezza dello 0.92, mentre il modello BNS su MICCAI ha avuto un'accuratezza dello 0.89.

I risultati mostrano chiaramente che FedAvg rappresenta l'approccio più equilibrato ed efficace tra quelli analizzati, soprattutto grazie alla sua capacità di mantenere elevate prestazioni su diversi dataset. Il metodo ha dimostrato una robustezza notevole, con un equilibrio tra accuratezza, precisione e recall, rendendolo adatto a diverse applicazioni di segmentazione di immagini mediche. FedMix, pur mostrando alcuni punti di forza, risulta meno affidabile a causa della variabilità delle prestazioni tra i dataset. Infine, FedST e FedRGD non sono raccomandabili per questo tipo di applicazioni a causa delle loro prestazioni insoddisfacenti.

Questa tesi ha messo in evidenza l'efficacia dell'apprendimento federato per la segmentazione di immagini mediche, identificando FedAvg come il metodo più robusto tra quelli analizzati. Tuttavia, rimangono numerose opportunità per migliorare ulteriormente le prestazioni e la generalizzabilità dei modelli di apprendimento federato. Per cominciare, è cruciale ottimizzare ulteriormente questi metodi. Infatti, anche se FedAvg ha mostrato buone prestazioni, c'è spazio

per miglioramenti attraverso l'implementazione di tecniche avanzate di ottimizzazione degli iperparametri di addestramento. Inoltre, combinare FedAvg con tecniche di personalizzazione del modello per specifici client potrebbe migliorare significativamente la generalizzabilità. Un'altra area di sviluppo riguarda l'integrazione di tecniche di data augmentation. Utilizzare strategie come il trasferimento di stile o la sintesi di dati potrebbe ampliare i dataset disponibili, migliorando così la robustezza e l'accuratezza dei modelli federati. Inoltre, sarebbe molto efficace trasformare questo lavoro in un problema multi-classe. Si potrebbero, infatti, impiegare delle tecniche particolari di colorazione dei nuclei cellulari per identificare in modo univoco le varie patologie, per fornire un ulteriore supporto ai medici competenti per quanto riguarda le diagnosi.

Per validare ulteriormente l'efficacia dell'apprendimento federato, sarà fondamentale condurre esperimenti su dataset più ampi e diversificati, inclusi dataset multimodali che combinano immagini mediche con ulteriori dati, come quelli clinici. Questo potrebbe fornire una valutazione più completa delle capacità e delle limitazioni dei modelli di apprendimento federato. Infine, oltre a promuovere collaborazioni interdisciplinari tra esperti di machine learning, professionisti del settore medico e ingegneri del software per l'applicazione pratica dell'apprendimento federato in campo medico, esplorare l'utilizzo dei vision transformer come un approccio futuro promettente potrebbe migliorare ulteriormente le prestazioni dei modelli di segmentazione. I vision transformer, noti per la loro capacità di catturare informazioni a lungo raggio nelle immagini, potrebbero integrarsi efficacemente con l'apprendimento federato per affrontare le sfide di generalizzazione e accuratezza nelle immagini mediche.

In conclusione, questa tesi ha dimostrato il potenziale dell'apprendimento federato nel contesto della segmentazione di immagini mediche e, più specificamente, nella segmentazione di nuclei cellulari da molteplici dataset eterogenei, fornendo diversi spunti per ottimizzare l'approccio e migliorare le performance ottenute attraverso gli sviluppi futuri menzionati.



# Bibliografia

- [1] T. Bdair, N. Navab, and S. Albarqouni. Fedperl: Semi-supervised peer learning for skin lesion classification. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 336–346, 2021.
- [2] P. Dey. Cancer nucleus: Morphology and beyond. *Diagn. Cytopathol.*, 2010.
- [3] Bao Tram Duong. Machine learning: Train loss vs test loss. <https://baotramduong.medium.com/machine-learning-train-loss-vs-test-loss-735ccd713291>. Accessed: 2024-06-14.
- [4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages xxx–xxx, Lille, France, July 2015.
- [5] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 249–256, 2010.
- [6] Hao Guan, Pew-Thian Yap, Andrea Bozoki, and Mingxia Liu. Federated learning for medical image analysis: A survey. *Computerized Medical Imaging and Graphics*, 84:101774, 2020.
- [7] Weiming Hua, Haoyuan Chen, Wanli Liu, Xiaoyan Li, Hongzan Sun, Xinyu Huang, Marcin Grzegorzek, and Chen Lia. A comparative study of gastric histopathology sub-size image classification: from linear regression to visual transformer. *ArXiv preprint arXiv:2205.12843*, 2022. Available at: <https://arxiv.org/abs/2205.12843>.

- [8] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [9] Naylor Peter Jack, Walter Thomas, Laé Marick, and Reyal Fabien. Tnbc: A dataset for nuclei segmentation. <https://zenodo.org/record/2579118>, 2018.
- [10] Adway Kanhere, Pranav Kulkarni, Paul H Yi, and Vishwa S Parekh. Privacy-preserving collaboration for multi-organ segmentation via federated learning from sites with partial labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2380–2387, 2024.
- [11] Hui Kong, Metin N. Gurcan, and Kamel Belkacem-Boussaid. Partitioning histopathological images: An integrated framework for supervised color-texture segmentation and cell splitting. *IEEE Trans. Medical Imaging*, 30(9):1661–1677, 2011.
- [12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017.
- [13] Neeraj Kumar, Ruchika Verma, Deepak Anand, Yanning Zhou, Omer Fahri Onder, Efstratios Tsougenis, Hao Chen, Pheng-Ann Heng, Jiahui Li, Zhiqiang Hu, Yunzhi Wang, Navid Alemi Koohbanani, Mostafa Jahanifar, Nedra Zamani Tajeddin, Ali Gooya, Nasir M. Rajpoot, Xuhua Ren, Sihang Zhou, Qian Wang, Dinggang Shen, Cheng-Kun Yang, Chi-Hung Weng, Wei-Hsiang Yu, Chao-Yuan Yeh, Shuang Yang, Shuoyu Xu, Pak-Hei Yeung, Peng Sun, Amirreza Mahbod, Gerald Schaefer, Isabella Ellinger, Rupert Ecker, Örjan Smedby, Chunliang Wang, Benjamin Chidester, That-Vinh Ton, Minh-Triet Tran, Jian Ma, Minh N. Do, Simon Graham, Quoc Dang Vu, Jin Tae Kwak, Akshaykumar Gunda, Raviteja Chunduri, Corey Hu, Xiaoyang Zhou, Dariush Lotfi, Reza Safdari, Antanas Kascenas, Alison O’Neil, Dennis Eschweiler, Johannes Stegmaier, Yanping Cui, Baocai Yin, Kailin Chen, Xinmei Tian, Philipp Grüning, Erhardt Barth, Elad Arbel, Itay Remer, Amir Ben-Dor, Ekaterina Sirazitdinova, Matthias Kohl, Stefan Braunewell, Yue-xiang Li, Xinpeng Xie, Linlin Shen, Jun Ma, Krishanu Das Baksi, Mohammad Azam Khan, Jaegul Choo, Adrián Colomer, Valery Naranjo, Linmin

Pei, Khan M. Iftekharuddin, Kaushiki Roy, Debotosh Bhattacharjee, Aníbal Pedraza, Maria Gloria Bueno, Sabarinathan Devanathan, Saravanan Radhakrishnan, Praveen Koduganty, Zihan Wu, Guanyu Cai, Xiaojie Liu, Yuqin Wang, and Amit Sethi. A multi-organ nucleus segmentation challenge. *IEEE Trans. Medical Imaging*, 39(5):1380–1391, 2020.

- [14] D. Liu, D. Zhang, Y. Song, F. Zhang, L. O'Donnell, H. Huang, M. Chen, and W. Cai. Unsupervised instance segmentation in microscopy images via panoptic domain adaptation and task re-weighting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020.
- [15] Xinlei Liu. Nuclei segmentation experiments demo. [https://github.com/lxmwust/Nuclei\\_Segmentation\\_Experiments\\_Demo](https://github.com/lxmwust/Nuclei_Segmentation_Experiments_Demo), 2023.
- [16] Zheng Liu, Jun Wang, et al. An accurate nuclei segmentation algorithm in pathological image based on deep semantic network. *ResearchGate*, 2019. Accessed: 2024-06-07.
- [17] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
- [18] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas. Communication-efficient learning of deep networks from decentralized data. *arXiv:1602.05629*, 2016.
- [19] Peter Naylor, Marick Lae, Fabien Reyal, and Thomas Walter. Nuclei segmentation in histopathology images using deep neural networks. In *14th IEEE International Symposium on Biomedical Imaging, ISBI 2017, Melbourne, Australia, April 18-21, 2017*, pages 933–936. IEEE, 2017.
- [20] Peter Naylor, Marick Lae, Fabien Reyal, and Thomas Walter. Segmentation of nuclei in histopathology images by deep regression of the distance map. *IEEE Trans. Medical Imaging*, 38(2):448–459, 2019.
- [21] Peter Naylor, Marick Laé, Fabien Reyal, and Thomas Walter. Segmentation of nuclei in histopathology images by deep regression of the distance map. <https://doi.org/10.21227/H26XOH>, May 2022.

- [22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.
- [23] Xiao Qi, Feng Xing, David J Foran, and Lin Yang. Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set. *IEEE Transactions on Biomedical Engineering*, 2012.
- [24] Xin Qi, Fuyong Xing, and David J. Foran. Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set. *IEEE Transactions on Biomedical Engineering*, 59(3):754–765, March 2012.
- [25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015.
- [26] Karin Stacke, Gabriel Eilertsen, Jonas Unger, and Claes Lundström. Measuring domain shift for deep learning in histopathology. *IEEE J. Biomed. Health Informatics*, 25(2):325–336, 2021.
- [27] V7 Labs. Intersection over union guide. <https://www.v7labs.com/blog/intersection-over-union-guide>. Accessed: 2024-06-14.
- [28] Ruchika Verma, Neeraj Kumar, Abhijeet Patil, Nikhil Cherian Kurian, Swapnil Rane, Simon Graham, Quoc Dang Vu, Mieke Zwager, Shan-E-Ahmed Raza, Nasir M. Rajpoot, Xiyi Wu, Huai Chen, Yijie Huang, Lisheng Wang, Hyun Jung, G. Thomas Brown, Yanling Liu, Shuolin Liu, Seyed Alireza Fate-mi Jahromi, Ali Asghar Khani, Ehsan Montahaei, Mahdieh Soleymani Bagh-shah, Hamid Behroozi, Pavel Semkin, Alexandre Rassadin, Prasad Dutande, Romil Lodaya, Ujjwal Baid, Bhakti Baheti, Sanjay N. Talbar, Amirreza Mah-bod, Rupert Ecker, Isabella Ellinger, Zhipeng Luo, Bin Dong, Zhengyu Xu, Yuehan Yao, Shuai Lv, Ming Feng, Kele Xu, Hasib Zunair, Abdessamad Ben

- Hamza, Steven M. Smiley, Tang-Kai Yin, Qi-Rui Fang, Shikhar Srivastava, Dwarikanath Mahapatra, Lubomira Trnavska, Hanyun Zhang, Priya Lakshmi Narayanan, Justin Law, Yinyin Yuan, Abhiroop Tejomay, Aditya Mitkari, Dinesh Koka, Vikas Ramachandra, Lata Kini, and Amit Sethi. Monusac2020: A multi-organ nuclei segmentation and classification challenge. *IEEE Trans. Medical Imaging*, 40(12):3413–3423, 2021.
- [29] Jason Wang, Luis Perez, et al. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, 11(2017):1–8, 2017.
- [30] Jeffry Wicaksana, Zengqiang Yan, Dong Zhang, Xijie Huang, Huimin Wu, Xin Yang, and Kwang-Ting Cheng. Fedmix: Mixed supervised federated learning for medical image segmentation. *IEEE*, 2023.
- [31] Y. Xu, P. Cao, Y. Kong, and Y. Wang. L\_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Proc. NeurIPS*, pages 6222–6233, 2019.
- [32] D. Yang et al. Federated semi-supervised learning for covid region segmentation in chest ct using multi-national data from china, italy, japan. *Med. Image Anal.*, 70:101992, May 2021. Art. no. 101992.
- [33] F. Yi, J. Huang, L. Yang, Y. Xie, and G. Xiao. Automatic extraction of cell nuclei from h&e-stained histopathological images. *J. Med. Imaging*, 2017.
- [34] Z. Zhang et al. Improving semi-supervised federated learning by reducing the gradient diversity of models. In *Proceedings of the International Conference on Big Data (ICBD)*, pages 1214–1225, 2021.



# Elenco delle figure

2.1	Esempio di immagine istopatologica catturata al microscopio [15]. . . . .	6
2.2	Esempio di coppia immagine-ground truth per la segmentazione di immagini istopatologiche [15]. . . . .	7
2.3	Esempio di immagine e relativa ground truth di due dataset diffe- renti: MICCAI e BNS [15]. . . . .	8
2.4	Organizzazione dell'apprendimento federato <sup>1</sup> . . . . .	9
3.1	Esempio di coppia immagine-ground truth MICCAI [15]. . . . .	12
3.2	Esempio di coppia immagine-ground truth BNS [15] . . . . .	13
3.3	Esempio di coppia immagine-ground truth MoNuSeg. Fonte: [13]. .	14
3.4	Esempio di coppia immagine-ground truth TNBC [9]. . . . .	14
3.5	Esempio di coppia immagine-ground truth MoNuSAC [28]. . . . .	15
3.6	Struttura di un sistema di FedAvg <sup>2</sup> . . . . .	18
3.7	Struttura di un sistema di FedMix [30]. . . . .	19
3.8	Sezione di codice per lo split delle immagini. La variabile TRAIN_RATIO è impostata a 0.8. La classe Cancer gestisce il preprocessing delle immagini e delle ground truth. . . . .	22
3.9	Sezione di codice per la data augmentation . . . . .	26
3.10	Esempi di trasformazioni applicate durante il processo di training. .	27
3.11	Implementazione in Python delle misure di performance <sup>3</sup> . . . . .	28



# Elenco delle tabelle

3.1	Dati delle immagini per dataset . . . . .	26
4.1	Risultati delle diverse prestazioni di FedAvg. Sono evidenziati in grassetto i due migliori risultati per misura di performance. . . . .	33
4.2	Risultati delle diverse prestazioni di FedMix. Sono evidenziati in grassetto i due migliori risultati per misura di performance. . . . .	34
4.3	Risultati delle diverse prestazioni di FedST. Sono evidenziati in grassetto i due migliori risultati per misura di performance. . . . .	34
4.4	Risultati delle diverse prestazioni di FedRGD. Sono evidenziati in grassetto i due migliori risultati per misura di performance. . . . .	34
4.5	Risultati delle diverse prestazioni di LL. Sono evidenziati in grassetto il miglior risultato per misura di performance. . . . .	34
4.6	Risultati delle prestazioni dei modelli FedAvg su dataset di test esterni. Sono evidenziati in grassetto i migliori risultati per misura di performance all'interno di ogni dataset. . . . .	35
4.7	Risultati delle prestazioni dei modelli FedMix su dataset di test esterni. Sono evidenziati in grassetto i migliori risultati per misura di performance all'interno di ogni dataset. . . . .	36
4.8	Risultati delle prestazioni dei modelli FedST su dataset di test esterni. . . . .	36
4.9	Risultati delle prestazioni dei modelli FedRGD su dataset di test esterni. . . . .	37
4.10	Risultati delle prestazioni dei modelli LL su dataset di test esterni. Sono evidenziati in grassetto i migliori risultati per misura di performance all'interno di ogni dataset. . . . .	37
4.11	Risultati delle diverse prestazioni di FedAvg. Sono evidenziati in grassetto i due migliori risultati per misura di performance. . . . .	38

4.12 Risultati delle diverse prestazioni di FedMix. Sono evidenziati in grassetto i due migliori risultati per misura di performance e dataset.	38
4.13 Risultati delle diverse prestazioni di FedAvg su dati esterni. Sono evidenziati in grassetto i due migliori risultati per misura di performance e dataset esterno. . . . .	38
4.14 Risultati delle diverse prestazioni di FedMix su dati esterni. Sono evidenziati in grassetto i due migliori risultati per misura di performance e dataset esterno. . . . .	39
4.15 Risultati delle diverse prestazioni di FedAvg (2 Clients) per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance. . . . .	39
4.16 Risultati delle diverse prestazioni di FedMix (2 Clients) per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance. . . . .	40
4.17 Risultati delle diverse configurazioni di FedST per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance. . . . .	40
4.18 Risultati delle diverse configurazioni di FedRGD per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance. . . . .	40
4.19 Risultati delle diverse configurazioni di LL per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance. . . . .	40
4.20 Risultati delle diverse prestazioni di FedAvg (3 Clients) per la sperimentazione cross-dataset. Sono evidenziati in grassetto i due risultati migliori per misura di performance. . . . .	41
4.21 Risultati delle diverse prestazioni di FedMix (3 Clients) per la sperimentazione cross-dataset. È evidenziato in grassetto il risultato migliore per misura di performance. . . . .	41
4.22 Migliori risultati per ogni metodo sul dataset MICCAI . . . . .	42
4.23 Migliori risultati per ogni metodo sul dataset BNS . . . . .	42
4.24 Migliori risultati per ogni metodo sul dataset MoNuSeg . . . . .	42
4.25 Migliori risultati per ogni metodo sul dataset TNBC . . . . .	43
4.26 Migliori risultati per ogni metodo sul dataset MoNuSAC . . . . .	43

# **Ringraziamenti**

*Desidero ringraziare tutti coloro che, nonostante le difficoltà e i momenti duri, sono stati sempre accanto a me e mi hanno supportato per superare ogni ostacolo. Ringrazio i miei genitori per avermi guidato e sostenuto in tutti questi anni. Ringrazio Mirco, Costantino, Pietro, Niccolò e tutti i colleghi con cui ho studiato e con cui mi sono divertito. Ringrazio il mio relatore e il mio correlatore per il sostegno ricevuto fino a questo giorno. Ringrazio infine tutti i professori del corso di laurea magistrale in informatica dell'Università di Cagliari per avermi insegnato tantissime cose di cui farò sicuramente buon uso negli anni a venire.*

*Auguro a tutti loro di realizzarsi nella vita e di ottenere tante soddisfazioni, così come io ne ho ottenuta una oggi, molto importante.*

