



ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Dipartimento di Informatica — Scienza e Ingegneria
Corso di Laurea Triennale in Informatica

ANALISI COMPARATIVA DI PIPELINE RADIOMICHE E LORO ESECUZIONE SU ARCHITETTURA RISC-V

Tesi di Laurea in Informatica

Relatore:

Chiar.mo Prof. Andrea Omicini

Presentata da:

Sara Marcheselli

Correlatori:

Dott. Francesco Giacomini

Dott.ssa Camilla Scapicchio

Anno Accademico 2025/2026

*Alla mia famiglia,
che non ha mai smesso di credere in me,
sostenendomi in ogni mia scelta
e permettendomi di realizzare i miei sogni.*

Introduzione

Negli ultimi anni la radiomica ha assunto un ruolo sempre più rilevante nell'analisi quantitativa delle immagini mediche, consentendo di trasformare esami diagnostici in un insieme di biomarcatori, detti feature radiomiche, utilizzabili per applicazioni di supporto alla diagnosi e alla medicina personalizzata. Tuttavia, l'estrazione di tali caratteristiche richiede pipeline software complesse e computazionalmente onerose, il cui comportamento dipende non solo dagli algoritmi implementati, ma anche dall'architettura hardware sulla quale vengono eseguite.

Parallelamente, l'architettura RISC-V sta emergendo come una valida alternativa alle tradizionali ISA (Instruction Set Architecture) proprietarie grazie alla sua natura aperta, modulare ed estensibile. Nonostante il crescente interesse verso questa piattaforma, risultano ancora limitati gli studi che ne analizzano il comportamento nell'ambito di applicazioni scientifiche reali e in particolare nel contesto della radiomica.

L'obiettivo di questa tesi è dimostrare che il confronto tra differenti implementazioni di pipeline radiomiche non può limitarsi alla sola analisi delle feature estratte, ma deve estendersi sia alla valutazione della loro riproducibilità e robustezza sia agli aspetti di portabilità ed efficienza computazionale. Attraverso il porting di due pipeline su architettura RISC-V, l'analisi dell'accordo tra le feature prodotte e la valutazione delle prestazioni su piattaforme RISC-V e x86 (in particolare la sua versione a 64 bit x86_64), viene mostrato come differenti scelte implementative possano influenzare significativamente i costi computazionali pur mantenendo risultati quantitativi sostanzialmente

equivalenti.

Per raggiungere gli obiettivi della tesi, il lavoro è stato articolato in una serie di fasi successive.

In primo luogo è stato analizzato il funzionamento di due differenti implementazioni software per l'estrazione di feature radiomiche. Sebbene entrambe perseguano il medesimo obiettivo e si basino sulla libreria PyRadiomics, esse adottano strategie differenti nella gestione dei dati, nelle operazioni di pre-processing e nell'organizzazione del flusso di elaborazione. Il confronto tra tali implementazioni costituisce un interessante caso di studio per valutare l'impatto delle scelte implementative sia sulle prestazioni computazionali sia sull'equivalenza dei risultati prodotti.

Successivamente è stata analizzata l'architettura RISC-V, approfondendone le principali caratteristiche progettuali e le differenze rispetto alle tradizionali piattaforme x86, al fine di comprenderne le possibili implicazioni nell'esecuzione di applicazioni scientifiche.

La fase successiva ha riguardato il porting delle due pipeline sulla piattaforma RISC-V. Questa attività ha richiesto l'identificazione delle incompatibilità presenti nelle librerie utilizzate, l'adattamento del software e la definizione delle modifiche necessarie per ottenere un'esecuzione stabile e corretta.

Una volta completato il porting, è stata condotta un'analisi della riproducibilità delle feature radiomiche estratte dalle due implementazioni. A tale scopo sono stati impiegati indicatori statistici di affidabilità e di accordo, in modo da quantificare il livello di equivalenza tra le misure prodotte dalle diverse pipeline e dalle differenti architetture hardware.

Infine, è stata effettuata una valutazione delle prestazioni computazionali delle pipeline, sia in modalità sequenziale sia in modalità parallela, confrontando l'architettura RISC-V con una piattaforma x86 di riferimento, al fine di analizzarne l'efficienza e l'idoneità all'esecuzione di carichi di lavoro radiomici.

Indice

Introduzione	ii
1 Radiomica e feature extraction	1
1.1 Introduzione alla radiomica	1
1.2 Pipeline radiomiche	3
1.2.1 Acquisizione delle immagini	4
1.2.2 Preprocessing e standardizzazione	5
1.2.3 Segmentazione della regione di interesse	5
1.2.4 Estrazione delle feature radiomiche	6
1.2.5 Standardizzazione e armonizzazione delle feature	7
1.2.6 Selezione e analisi delle feature più rilevanti	8
2 Architettura RISC-V	9
2.1 Introduzione alle ISA	9
2.2 Architettura RISC-V	10
2.3 Organizzazione dell'ISA	11
2.4 Confronto con architettura x86	12
2.5 Piattaforme hardware utilizzate	13
3 Pipeline radiomiche per estrazione di feature	17
3.1 Prima Pipeline	19
3.2 Seconda Pipeline	20

4	Porting e adattamento delle pipeline su RISC-V	23
4.1	Obiettivi del porting	23
4.2	Problematiche di compatibilità e adattamenti per RISC-V . .	24
4.2.1	Compilazione conservativa di NumPy	25
4.2.2	Disabilitazione dell'estensione vettoriale RVV	25
4.2.3	Correzione di un'incompatibilità in PyRadiomics . . .	26
5	Parallelizzazione delle pipeline di estrazione delle feature	29
5.1	Motivazioni della parallelizzazione	29
5.2	Strategia di parallelizzazione adottata	30
5.3	Implementazione	30
5.4	Considerazioni sull'architettura RISC-V	31
6	Valutazione comparativa delle pipeline radiomiche	33
6.1	Confronto di riproducibilità delle feature radiomiche	33
6.1.1	Analisi di riproducibilità fra le pipeline	34
6.1.2	Analisi di riproducibilità fra le architetture	38
6.2	Confronto delle prestazioni computazionali	38
6.2.1	Senza parallelizzazione	39
6.2.2	Con parallelizzazione	41
	Bibliografia	50

Elenco delle figure

1.1	Schema generale di una pipeline radiomica. Fonte: [7].	3
3.1	Esempio di una scansione MRI appartenente al dataset utilizzato, della corrispondente maschera della regione di interesse (ROI) e della loro sovrapposizione.	19
6.1	Distribuzione dei coefficienti ICC e CV per classe di feature. .	34
6.2	Grafico a dispersione della correlazione tra l'indice di affidabilità $ICC(A, 1)$ e il Coefficiente di Variazione medio.	35
6.3	Analisi dettagliata delle performance di parallelizzazione confrontando le due piattaforme hardware (x86 e RISC-V) e le due implementazioni software (Pipeline 1 e Pipeline 2).	42

Elenco delle tabelle

1.1	Classificazione delle feature radiomiche, con indicazione del numero di feature per ciascuna famiglia.	7
2.1	Specifiche della piattaforma sperimentale RISC-V utilizzata . .	14
2.2	Specifiche della piattaforma sperimentale x86 utilizzata	14
6.1	Analisi di affidabilità (ICC) e accordo relativo (CV) per le feature morfologiche (classe Shape).	36
6.2	Confronto delle metriche prestazionali tra le esecuzioni sequenziali delle pipeline su architetture x86 e RISC-V.	39
6.3	Confronto dei tempi totali di esecuzione (Wall-clock) tra architetture x86 e RISC-V per le Pipeline 1 e 2 al variare del numero di processi.	41

Capitolo 1

Radiomica e feature extraction

1.1 Introduzione alla radiomica

Negli ultimi decenni le tecniche di imaging medico hanno assunto un ruolo centrale nella diagnosi, nella pianificazione terapeutica e nel monitoraggio delle patologie. Modalità di acquisizione come la Tomografia Computerizzata (Computed Tomography, CT), la Risonanza Magnetica (Magnetic Resonance Imaging, MRI) e la Tomografia a Emissione di Positroni (Positron Emission Tomography, PET) consentono di ottenere immagini dettagliate delle strutture anatomiche e dei processi fisiologici del corpo umano. Tradizionalmente tali immagini vengono interpretate da specialisti attraverso una valutazione qualitativa basata sull'esperienza clinica e sull'osservazione visiva.

Tuttavia, una parte significativa delle informazioni contenute nelle immagini mediche non è direttamente percepibile dall'occhio umano. Differenze minime nella distribuzione delle intensità, nella struttura dei tessuti o nella presenza di pattern complessi possono essere difficili da individuare mediante la sola osservazione visiva. Per questo motivo, negli ultimi anni si è sviluppata la radiomica, una disciplina che si propone di trasformare le immagini mediche in un grande numero di dati quantitativi utilizzabili per analisi statistiche e modelli di apprendimento automatico, contribuendo allo sviluppo della medicina personalizzata.

La radiomica si basa sull'estrazione sistematica di numerosi descrittori quantitativi dalle immagini mediche. Questi descrittori, chiamati "feature radiomiche", rappresentano proprietà misurabili dell'immagine e possono essere utilizzati per caratterizzare tessuti, organi o lesioni in maniera oggettiva e riproducibile.

L'interesse verso la radiomica è cresciuto rapidamente grazie alla crescente disponibilità di immagini digitali e all'aumento della capacità computazionale dei sistemi di elaborazione. Le moderne infrastrutture informatiche consentono infatti di processare grandi volumi di dati e di estrarre centinaia o migliaia di feature da ciascun esame diagnostico.

Per garantire la robustezza delle analisi è necessario seguire una pipeline ben definita che comprende diverse fasi successive: acquisizione delle immagini, preprocessing e standardizzazione, segmentazione della regione di interesse, estrazione delle feature e selezione delle caratteristiche più rilevanti. Ogni fase contribuisce alla qualità finale dei dati prodotti e può influenzare le prestazioni degli algoritmi sviluppati a partire dalle feature estratte.

Dal punto di vista informatico, la radiomica rappresenta un interessante caso di studio per l'analisi di pipeline computazionali complesse. L'estrazione delle feature richiede infatti l'esecuzione di numerosi algoritmi di elaborazione delle immagini e di calcolo statistico che possono risultare particolarmente onerosi dal punto di vista computazionale. L'efficienza delle implementazioni software e l'architettura hardware utilizzata possono quindi avere un impatto significativo sui tempi di elaborazione e sulle prestazioni complessive del sistema.

Nel contesto di questa tesi, l'attenzione sarà focalizzata proprio sugli aspetti computazionali delle pipeline radiomiche. In particolare verranno analizzate due diverse implementazioni software per l'estrazione di feature radiomiche da immagini mediche e ne verranno confrontate le prestazioni su architetture hardware differenti, con particolare attenzione alla piattaforma RISC-V. Prima di affrontare tali aspetti, nei paragrafi successivi verranno descritti i principali elementi che compongono una pipeline radiomica e le tecni-

che utilizzate per l'estrazione delle informazioni quantitative dalle immagini mediche.

1.2 Pipeline radiomiche

L'obiettivo principale della pipeline è estrarre informazioni numeriche dalle immagini diagnostiche in modo standardizzato, riproducibile e indipendente dall'interpretazione soggettiva dell'operatore. Ogni fase della pipeline svolge un ruolo specifico e contribuisce alla qualità delle feature finali.

Sebbene possano esistere leggere variazioni a seconda dell'applicazione clinica o del software utilizzato, una pipeline radiomica è generalmente composta dalle seguenti fasi:

1. Acquisizione delle immagini mediche;
2. Preprocessing e standardizzazione;
3. Segmentazione della regione di interesse (ROI);
4. Estrazione delle feature radiomiche;
5. Standardizzazione e armonizzazione delle feature;
6. Selezione e analisi delle feature più rilevanti.

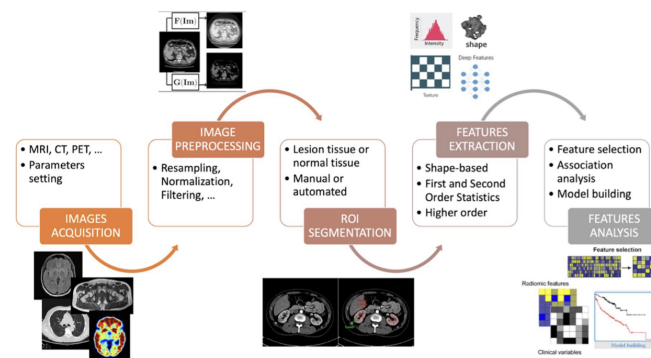


Figura 1.1: Schema generale di una pipeline radiomica. Fonte: [7].

L'accuratezza e l'affidabilità dell'intero processo dipendono dalla corretta esecuzione di ciascuna fase. Errori introdotti durante l'acquisizione delle immagini, nella segmentazione della lesione o nella normalizzazione dei dati possono propagarsi lungo tutta la pipeline e influenzare significativamente le prestazioni dei modelli sviluppati successivamente.

Per questo motivo la pipeline radiomica viene considerata un processo integrato in cui ogni passaggio contribuisce alla produzione di biomarcatori quantitativi robusti e riproducibili.

1.2.1 Acquisizione delle immagini

La prima fase della pipeline consiste nell'acquisizione delle immagini mediche che verranno successivamente analizzate. Le immagini possono essere ottenute mediante differenti modalità diagnostiche, tra cui Tomografia Computerizzata (CT), Risonanza Magnetica (MRI) e Tomografia a Emissione di Positroni (PET). A differenza della radiografia convenzionale, tali tecniche acquisiscono una sequenza di immagini bidimensionali (slice) lungo un determinato asse anatomico, consentendo la ricostruzione di un volume tridimensionale del paziente. Tale rappresentazione volumetrica è costituita da elementi discreti denominati voxel, equivalenti tridimensionali dei pixel.

La qualità delle immagini acquisite influenza direttamente le feature radiomiche che verranno estratte nelle fasi successive. Uno dei principali problemi della radiomica è la presenza di differenze tra scanner e protocolli di acquisizione appartenenti a differenti centri clinici. Tali differenze possono introdurre variazioni non legate alle caratteristiche biologiche del paziente ma esclusivamente alle modalità di acquisizione delle immagini. Per questo motivo sono necessarie procedure di standardizzazione che verranno descritte nella sezione successiva.

1.2.2 Preprocessing e standardizzazione

Le immagini provenienti da differenti dispositivi diagnostici presentano spesso caratteristiche non omogenee. Prima dell'estrazione delle feature è quindi necessario applicare una serie di operazioni di preprocessing finalizzate alla riduzione delle variabilità tecniche.

Tra le operazioni più comuni rientrano il resampling dei voxel, la normalizzazione delle intensità e la discretizzazione dei livelli di grigio.

Il resampling consente di uniformare la dimensione dei voxel presenti nelle immagini, garantendo che le feature vengano calcolate su dati caratterizzati dalla stessa risoluzione spaziale.

La normalizzazione delle intensità agisce sui valori dei voxel, riportandoli a una distribuzione o a un intervallo di valori comune, al fine di limitare l'influenza delle variazioni introdotte dal processo di acquisizione e migliorare la robustezza delle feature estratte.

Infine, la discretizzazione dei livelli di grigio consiste nella riduzione del numero di intensità presenti nell'immagine attraverso il raggruppamento in intervalli predefiniti. Questa operazione migliora la stabilità di molte feature di texture (introdotte nella Sezione 1.2.4) e riduce l'impatto del rumore.

1.2.3 Segmentazione della regione di interesse

La segmentazione rappresenta una delle fasi più importanti della pipeline radiomica. Essa consiste nell'identificazione della regione dell'immagine contenente il tessuto o la lesione che si desidera analizzare.

La regione selezionata prende il nome di Region of Interest (ROI). Le feature radiomiche vengono calcolate esclusivamente all'interno della ROI; di conseguenza la qualità della segmentazione influenza direttamente i risultati finali.

La segmentazione può essere effettuata manualmente da un radiologo esperto, mediante approcci semi-automatici oppure attraverso algoritmi completamente automatici basati su tecniche di computer vision e deep learning.

La segmentazione manuale garantisce generalmente un'elevata accuratezza ma richiede tempi significativi ed è soggetta a variabilità intra-operatore e inter-operatore. Per questo motivo negli ultimi anni sono stati sviluppati numerosi algoritmi automatici, tra cui metodi di thresholding, clustering e reti neurali convoluzionali.

1.2.4 Estrazione delle feature radiomiche

Dopo aver individuato la regione di interesse, la pipeline procede con l'estrazione delle feature radiomiche. Questa fase trasforma la ROI in un insieme di descrittori numerici che caratterizzano quantitativamente il tessuto analizzato.

Le feature radiomiche possono essere suddivise in tre principali categorie.

Le feature di primo ordine descrivono la distribuzione statistica dei valori di intensità dei voxel appartenenti alla ROI, senza considerare le relazioni spaziali tra essi. Alcuni esempi sono media, mediana, deviazione standard ed entropia.

Le feature di forma descrivono le proprietà geometriche e morfologiche della regione segmentata, indipendentemente dai valori di intensità dei voxel, come volume, area, compattezza e sfericità.

Le feature di texture di secondo ordine descrivono l'organizzazione spaziale dei livelli di intensità all'interno della ROI e consentono di quantificare proprietà quali eterogeneità, uniformità e complessità del tessuto. Esse vengono calcolate a partire da matrici che modellano differenti relazioni spaziali tra i voxel, tra cui la Gray Level Co-occurrence Matrix (GLCM), la Gray Level Run Length Matrix (GLRLM) e la Gray Level Size Zone Matrix (GLSZM).

L'estrazione delle feature viene spesso effettuata mediante librerie software dedicate. Una delle soluzioni più diffuse è PyRadiomics [1][2], una libreria open-source sviluppata in Python, alla base delle pipeline che verranno prese in esame in seguito. L'implementazione di PyRadiomics si basa su diverse librerie software specializzate. In particolare, NumPy [3] è utilizza-

ta per le operazioni numeriche sugli array multidimensionali, SimpleITK [4] per la lettura, la scrittura e la manipolazione delle immagini mediche, mentre PyWavelets [5] è impiegata per il calcolo delle feature ottenute mediante trasformata wavelet.

Classe	Famiglia di Feature	Numero di Feature
First Order	First Order Statistics	19
Shape	Shape-based (3D)	16
	Shape-based (2D)	10
Second Order	Gray Level Co-occurrence Matrix (GLCM)	24
	Gray Level Run Length Matrix (GLRLM)	16
	Gray Level Size Zone Matrix (GLSZM)	16
	Neighbouring Gray Tone Difference Matrix (NGTDM)	5
	Gray Level Dependence Matrix (GLDM)	14
Totale		120

Tabella 1.1: Classificazione delle feature radiomiche, con indicazione del numero di feature per ciascuna famiglia.

1.2.5 Standardizzazione e armonizzazione delle feature

Affinché le feature radiomiche possano essere utilizzate in studi multicentrici o confrontate tra differenti software, è necessario garantire adeguati livelli di riproducibilità.

La standardizzazione delle feature viene promossa dall’Image Biomarker Standardization Initiative (IBSI), un progetto internazionale che definisce procedure comuni per il calcolo delle feature radiomiche.

Oltre alla standardizzazione, è spesso necessario applicare tecniche di armonizzazione per ridurre gli effetti introdotti da differenti scanner o protocolli di acquisizione. Tra i metodi maggiormente utilizzati vi è ComBat, una tecnica statistica che consente di ridurre le differenze sistematiche mantenendo inalterata l’informazione biologicamente rilevante.

1.2.6 Selezione e analisi delle feature più rilevanti

L'estrazione delle feature radiomiche produce generalmente un elevato numero di descrittori quantitativi per ciascuna immagine. Tuttavia, i dataset clinici sono spesso caratterizzati da un numero relativamente limitato di pazienti rispetto al numero di feature disponibili. Questa condizione, nota come problema dell'alta dimensionalità, può favorire fenomeni di overfitting e compromettere la capacità di generalizzazione dei modelli predittivi.

Per questo motivo, una delle prime fasi dell'analisi consiste nella selezione delle feature più informative, con l'obiettivo di eliminare variabili ridondanti o poco rilevanti e ridurre la dimensionalità del problema.

Le feature selezionate vengono successivamente impiegate per lo sviluppo di modelli statistici e di machine learning, finalizzati a compiti quali classificazione, regressione o stratificazione dei pazienti. Tali modelli mirano a individuare relazioni tra le caratteristiche radiomiche e specifici endpoint clinici, come la diagnosi, la prognosi, la risposta al trattamento o la probabilità di recidiva della malattia.

Capitolo 2

Architettura RISC-V

Al fine di contestualizzare le attività descritte nei capitoli successivi, questo capitolo presenta una panoramica dell'architettura RISC-V, illustrandone i principi fondamentali e le caratteristiche che la rendono una piattaforma particolarmente interessante per l'esecuzione di applicazioni ad alte prestazioni.

2.1 Introduzione alle ISA

L'Instruction Set Architecture (ISA) rappresenta l'interfaccia fondamentale tra hardware e software. Essa definisce il set di istruzioni che un processore è in grado di eseguire, il formato dei dati, l'organizzazione dei registri e le modalità di accesso alla memoria. Attraverso l'ISA, i programmi possono interagire con l'hardware in modo standardizzato, indipendentemente dai dettagli implementativi della microarchitettura sottostante.

Dal punto di vista del software, l'ISA costituisce il livello di astrazione che garantisce la compatibilità tra programmi e processori. Due CPU differenti possono infatti eseguire lo stesso programma purché implementino la medesima ISA, anche se utilizzano soluzioni hardware completamente diverse.

Nel corso degli anni sono state sviluppate numerose ISA, ciascuna caratterizzata da specifiche scelte progettuali. Tra le più diffuse si possono citare x86, dominante nel mercato desktop e server, ARM, ampiamente utilizzata nei dispositivi mobili, e RISC-V, una soluzione open-source che negli ultimi anni ha suscitato notevole interesse sia in ambito industriale sia nella ricerca accademica.

La progettazione di una ISA influenza direttamente aspetti quali prestazioni, consumi energetici, complessità hardware e possibilità di espansione futura. Per questo motivo l'architettura del processore rappresenta un elemento fondamentale nello sviluppo di sistemi informatici moderni.

2.2 Architettura RISC-V

RISC-V è una Instruction Set Architecture open-source sviluppata presso l'Università della California, Berkeley, con l'obiettivo di fornire una piattaforma moderna, flessibile e liberamente utilizzabile per la progettazione di processori.

A differenza delle ISA proprietarie, l'utilizzo di RISC-V non richiede licenze commerciali o il pagamento di royalty. Questa caratteristica ha favorito la rapida diffusione dell'architettura sia in ambito accademico sia industriale, consentendo a università, aziende e centri di ricerca di sviluppare implementazioni personalizzate senza particolari vincoli.

L'architettura segue i principi della filosofia RISC (Reduced Instruction Set Computer), che si basa sull'idea di semplificare il set di istruzioni del processore, limitandolo a un numero ridotto di operazioni semplici e di lunghezza fissa. A differenza delle architetture CISC (Complex Instruction Set Computer), che implementano istruzioni complesse in grado di svolgere più operazioni contemporaneamente, un processore RISC privilegia istruzioni elementari che vengono eseguite molto rapidamente. Sebbene operazioni complesse richiedano un numero maggiore di istruzioni, la semplicità dell'hard-

ware permette di ridurre la complessità progettuale e di migliorare l'efficienza dell'esecuzione.

Negli ultimi anni RISC-V ha trovato applicazione in numerosi settori, tra cui sistemi embedded, Internet of Things, acceleratori per l'intelligenza artificiale e piattaforme per il calcolo scientifico. La crescente maturità dell'ecosistema software e hardware rende oggi possibile l'esecuzione di applicazioni complesse, incluse quelle basate su linguaggi ad alto livello come Python e librerie scientifiche utilizzate nell'ambito della radiomica.

La combinazione di apertura, modularità e supporto software rappresenta uno dei principali motivi che hanno portato alla scelta di RISC-V come piattaforma di riferimento per il presente lavoro.

2.3 Organizzazione dell'ISA

Una delle caratteristiche più innovative di RISC-V è la sua organizzazione modulare. L'architettura è infatti composta da una base minima di istruzioni, alla quale possono essere aggiunte estensioni opzionali in funzione delle esigenze applicative.

La base dell'ISA è rappresentata dalle versioni RV32I e RV64I, progettate rispettivamente per sistemi a 32 e 64 bit. Esse contengono il set essenziale di istruzioni necessario all'esecuzione di programmi generici.

A partire da questa base è possibile integrare numerose estensioni standard. Tra le più importanti si possono citare:

- M, che introduce operazioni di moltiplicazione e divisione;
- A, dedicata alle operazioni atomiche e alla programmazione concorrente;
- F, che aggiunge il supporto floating-point a singola precisione;
- D, che estende il supporto floating-point alla doppia precisione;

- C, che introduce istruzioni compresse per ridurre l'occupazione in memoria;
- V, che implementa operazioni vettoriali per l'elaborazione parallela di grandi quantità di dati.

Questa struttura modulare consente di adattare il processore alle specifiche necessità applicative, evitando l'inclusione di componenti non necessari e migliorando l'efficienza complessiva del sistema.

Particolarmente interessante per il calcolo scientifico è l'estensione vettoriale, che permette di elaborare simultaneamente insiemi di dati mediante una singola istruzione. Tale caratteristica risulta potenzialmente vantaggiosa per applicazioni di elaborazione delle immagini e per il calcolo delle feature radiomiche, che coinvolgono grandi matrici di dati numerici.

2.4 Confronto con architettura x86

Per comprendere il contesto nel quale si inserisce l'analisi svolta in questa tesi, è utile confrontare l'architettura RISC-V con x86, attualmente la piattaforma dominante nel settore dei personal computer e dei server.

L'architettura x86 ha origini negli anni Settanta ed è stata sviluppata nel corso dei decenni mantenendo un elevato livello di retrocompatibilità. Questo ha portato all'introduzione di numerose estensioni e all'aumento della complessità dell'Instruction Set Architecture. Le moderne CPU x86 implementano infatti un ampio insieme di istruzioni specializzate e utilizzano sofisticati meccanismi interni per massimizzare le prestazioni.

RISC-V, al contrario, privilegia semplicità, modularità e facilità di implementazione. La presenza di una base minima di istruzioni e di estensioni opzionali permette di costruire processori adattati a differenti scenari applicativi senza introdurre complessità non necessarie.

Dal punto di vista delle prestazioni, le moderne CPU x86 rappresentano attualmente una soluzione estremamente matura e ottimizzata. Tuttavia,

la crescente diffusione di RISC-V ha stimolato numerose attività di ricerca volte a valutare il comportamento di applicazioni reali su questa nuova architettura.

Un ulteriore elemento distintivo riguarda il modello di sviluppo. Mentre x86 è una tecnologia proprietaria controllata da un numero limitato di produttori, RISC-V è una specifica aperta che consente a qualsiasi organizzazione di sviluppare implementazioni compatibili.

Nel contesto del calcolo scientifico, entrambe le architetture offrono strumenti per accelerare l'elaborazione numerica. Poiché le pipeline radiomiche prevedono l'elaborazione di grandi quantità di dati e il calcolo di numerose statistiche sulle immagini mediche, esse rappresentano un interessante caso di studio per confrontare il comportamento di differenti architetture hardware. L'analisi sperimentale presentata nei capitoli successivi si propone quindi di valutare la portabilità e l'esecuzione di tali workload su piattaforma RISC-V, confrontandone le prestazioni con quelle ottenute su sistemi tradizionali basati su architettura x86.

2.5 Piattaforme hardware utilizzate

Le attività sperimentali descritte in questa tesi sono state condotte utilizzando due piattaforme hardware: una basata su architettura RISC-V e una basata su x86, impiegata come baseline prestazionale di confronto. Lo scopo dell'analisi è valutare il comportamento di pipeline radiomiche reali durante l'esecuzione su un sistema caratterizzato da una ISA differente rispetto alle tradizionali piattaforme x86 comunemente utilizzate in ambito scientifico.

Per garantire la riproducibilità degli esperimenti è necessario descrivere le principali caratteristiche hardware e software dei sistemi utilizzati. In particolare, risultano rilevanti il modello del processore, il numero di core disponibili, la quantità di memoria RAM installata, il sistema operativo e le versioni degli strumenti software impiegati per l'esecuzione delle pipeline.

Componente	Specifica
Architettura	RISC-V RV64
Numero di core	64
Memoria RAM	128 GB
Sistema operativo	Fedora Linux 38
Versione Python	Python 3.11.5
Librerie principali	PyRadiomics, NumPy, Nibabel, SimpleITK

Tabella 2.1: Specifiche della piattaforma sperimentale RISC-V utilizzata

Componente	Specifica
Architettura	x86_64
Numero di core	24
Memoria RAM	64 GB
Sistema operativo	AlmaLinux Linux 9.8
Versione Python	Python 3.9.25
Librerie principali	PyRadiomics, NumPy, Nibabel, SimpleITK

Tabella 2.2: Specifiche della piattaforma sperimentale x86 utilizzata

L'ambiente software in entrambe le architetture è stato configurato appositamente per consentire l'esecuzione completa delle pipeline radiomiche considerate nello studio. In particolare, è stato creato un ambiente virtuale Python dedicato, all'interno del quale sono state installate manualmente tutte le librerie necessarie, tra cui PyRadiomics, NumPy, Nibabel [6] e SimpleITK, insieme alle rispettive dipendenze. Questa scelta ha permesso di isolare l'ambiente di esecuzione dal sistema operativo, garantendo una configurazione controllata e facilmente riproducibile.

Le misure prestazionali riportate nei capitoli successivi sono state ottenute utilizzando queste due configurazioni hardware e software, mantenendo costanti le condizioni di esecuzione per garantire la confrontabilità dei risul-

tati. La descrizione dettagliata delle metriche raccolte verrà presentata nel capitolo dedicato all'analisi delle prestazioni.

Capitolo 3

Pipeline radiomiche per estrazione di feature

Le attività sperimentali descritte in questa tesi sono state condotte utilizzando due differenti pipeline software per l'estrazione di feature radiomiche da immagini di risonanza magnetica mammaria. Entrambe le implementazioni derivano da lavori precedenti sviluppati nell'ambito della ricerca in radiomica; tuttavia, sono state successivamente modificate e adattate nell'ambito del presente lavoro per consentirne l'esecuzione su un dataset costituito da immagini di risonanza magnetica relative a 93 pazienti. Inoltre, sono stati effettuati interventi specifici per garantirne la compatibilità con la piattaforma RISC-V, per introdurre una versione parallelizzata delle pipeline oggetto di studio e per integrare funzionalità di monitoraggio delle prestazioni, necessarie alla raccolta dei tempi di esecuzione durante gli esperimenti. Infine, è stata eseguita un'analisi comparativa delle feature estratte dalle due pipeline, con l'obiettivo di valutarne il grado di concordanza.

La prima pipeline analizzata è un'implementazione di un workflow radiomico standard sviluppata nell'ambito delle attività di ricerca del gruppo di lavoro.

La seconda pipeline deriva invece dal lavoro presentato da Morcaldi et al. [8] nello studio dedicato alla classificazione di lesioni mammarie mediante

analisi radiomica e algoritmi di machine learning.

Sebbene entrambe le pipeline utilizzino PyRadiomics come strumento principale per il calcolo delle feature, esse adottano approcci differenti nella gestione dei dati di ingresso, nelle operazioni di preprocessing e nell'organizzazione complessiva del flusso di elaborazione. Per questo motivo rappresentano un interessante caso di studio per valutare il comportamento di applicazioni radiomiche reali su architettura RISC-V.

Il dataset di risonanza magnetica mammaria utilizzato per le valutazioni sperimentali deriva dall'archivio pubblico *Advanced MRI Breast Lesions* (AMBL), ospitato su *The Cancer Imaging Archive* (TCIA) [10].

Le immagini cliniche originali sono archiviate nel formato DICOM, lo standard utilizzato in ambito medico per la memorizzazione delle immagini diagnostiche e delle relative informazioni sul paziente e sull'acquisizione. In questo lavoro, tali immagini sono state convertite nel formato NIfTI, uno standard ampiamente utilizzato nell'analisi di immagini mediche e nelle applicazioni di radiomica.

La conversione è stata effettuata perché il formato NIfTI semplifica l'elaborazione dei dati. Nel formato DICOM, infatti, un esame tridimensionale è costituito da un insieme di file, ciascuno corrispondente a una singola sezione bidimensionale dell'acquisizione (slice). Nel formato NIfTI, invece, tutte le slice vengono riunite in un unico file che rappresenta direttamente l'intero volume tridimensionale. Le informazioni necessarie per ricostruire correttamente la posizione e l'orientamento del volume sono memorizzate nell'header del file, rendendo il formato più pratico ed efficiente per le successive fasi di analisi.

Nello specifico, sono state utilizzate immagini di risonanza magnetica tridimensionali pesate in T2, una particolare sequenza di acquisizione che produce un contrasto in cui i fluidi e molti tessuti patologici appaiono più luminosi (iperintensi). Tali immagini presentano una risoluzione planare fissa di 512×512 pixel per ciascuna slice e una profondità longitudinale variabile tra 108 e 116 slice. Il volume tipico per paziente è quindi rappresentabile da

una griglia indicativa di $512 \times 512 \times 116$ voxel.

Oltre a queste immagini, sono state utilizzate anche le corrispondenti maschere di segmentazione tridimensionali delle lesioni, che identificano la regione di interesse (ROI) associata a ciascuna lesione e consentono di delimitare l'area da cui estrarre i parametri radiomici.

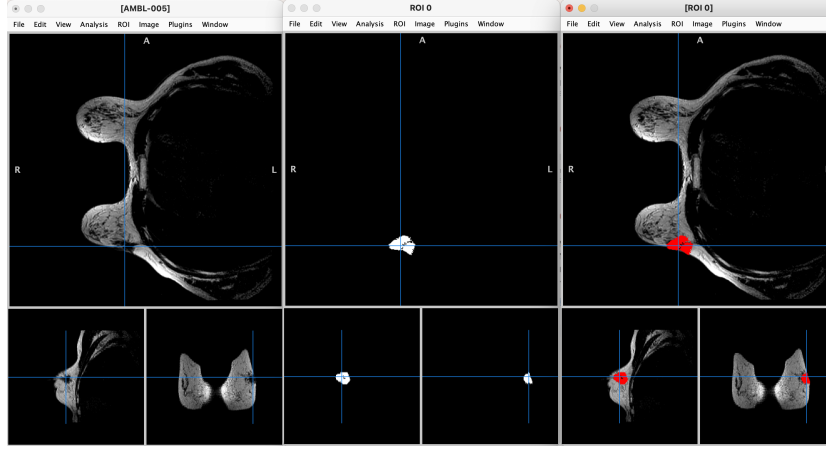


Figura 3.1: Esempio di una scansione MRI appartenente al dataset utilizzato, della corrispondente maschera della regione di interesse (ROI) e della loro sovrapposizione.

3.1 Prima Pipeline

La prima pipeline analizzata implementa un flusso di lavoro radiomico standard, caratterizzato da un approccio diretto che delega l'intero processo di elaborazione alla libreria PyRadiomics. Il calcolo delle feature viene guidato da un file di configurazione in formato YAML, il quale definisce i parametri essenziali per l'analisi, inclusi i criteri di discretizzazione delle intensità, le classi di feature da calcolare ed eventuali filtri o operazioni di preprocessing da applicare.

Il flusso di esecuzione inizia con l'esplorazione della cartella contenente i dati dei pazienti. Per ciascun soggetto vengono individuate l'immagine di risonanza magnetica e le corrispondenti maschere di segmentazione. La pipe-

line è in grado di gestire la presenza di più regioni di interesse (ROI) associate allo stesso paziente, effettuando l'estrazione delle feature separatamente per ogni lesione identificata.

Una volta individuata una coppia immagine-maschera, la pipeline richiama direttamente le funzionalità offerte da PyRadiomics. La libreria si occupa della lettura dei file NIfTI, della verifica della coerenza geometrica tra immagine e segmentazione e del calcolo delle feature radiomiche. Questo approccio consente di delegare alla libreria la maggior parte delle operazioni di basso livello, mantenendo il codice relativamente semplice e facilmente manutenibile.

Dal punto di vista architetturale, questa pipeline rappresenta una soluzione relativamente diretta, in cui la maggior parte delle operazioni viene delegata a PyRadiomics. Tale caratteristica la rende un utile riferimento per valutare le prestazioni di un workflow radiomico standard su piattaforma RISC-V, fornendo una base di confronto rispetto a implementazioni più complesse e maggiormente personalizzate.

3.2 Seconda Pipeline

A differenza della pipeline descritta nella sezione precedente, questa seconda implementazione non si limita a delegare l'intero processo di elaborazione a PyRadiomics, ma introduce una fase preliminare di gestione e trasformazione dei dati. L'obiettivo è ottenere un controllo più diretto sulle immagini e sulle informazioni geometriche associate alle lesioni prima dell'estrazione delle feature.

Il processo inizia con il caricamento delle immagini di risonanza magnetica e delle relative maschere di segmentazione tramite la libreria Nibabel. Nello specifico, i file vengono convertiti in array tridimensionali NumPy, consentendo di accedere direttamente ai valori dei voxel e di applicare eventuali operazioni di manipolazione sui dati. Questa scelta offre una maggiore flessibilità rispetto all'utilizzo diretto dei file NIfTI da parte di PyRadiomics.

Una caratteristica distintiva della pipeline riguarda la gestione dell'orientamento delle immagini. Durante le fasi di preprocessing vengono infatti eseguite operazioni di rotazione e riflessione degli array, necessarie per garantire la corretta corrispondenza spaziale tra immagini e segmentazioni. Tale passaggio consente di uniformare la rappresentazione dei dati prima della successiva fase di analisi.

Un ulteriore elemento di differenziazione è rappresentato dalla gestione esplicita delle informazioni geometriche. La pipeline recupera infatti parametri quali pixel spacing e slice thickness da un file CSV esterno contenente i metadati dei pazienti, a differenza della prima pipeline che, non passando per la conversione esplicita in NumPy, recupera automaticamente con PyRadiomics la dimensione originale del voxel contenuta nell'header NIfTI di ciascun paziente. Queste informazioni vengono successivamente associate agli array NumPy attraverso gli oggetti SimpleITK utilizzati durante l'estrazione delle feature. In questo modo viene preservata la corretta interpretazione spaziale delle lesioni, aspetto fondamentale per il calcolo di numerose feature radiomiche.

Terminata la fase di preprocessing, gli array NumPy vengono convertiti in immagini SimpleITK e forniti a PyRadiomics per l'estrazione delle feature. Anche in questo caso vengono calcolate diverse categorie di descrittori radiomici, tra cui feature morfologiche, statistiche del primo ordine e caratteristiche di texture basate sulle matrici GLCM, GLRLM, GLSZM, GLDM e NGTDM.

Dal punto di vista architetturale, questa soluzione risulta più articolata rispetto alla pipeline precedentemente analizzata. La presenza di operazioni aggiuntive di caricamento, trasformazione e gestione dei metadati introduce un maggiore carico computazionale, ma offre allo stesso tempo una maggiore flessibilità nel trattamento dei dati. Proprio queste differenze rendono particolarmente interessante il confronto prestazionale tra le due implementazioni, che verrà presentato in seguito.

Capitolo 4

Porting e adattamento delle pipeline su RISC-V

4.1 Obiettivi del porting

L'obiettivo principale del presente lavoro è valutare la fattibilità dell'esecuzione di pipeline radiomiche reali su una piattaforma RISC-V e analizzarne il comportamento dal punto di vista prestazionale.

A differenza di numerosi studi presenti in letteratura, che si concentrano principalmente sull'accuratezza diagnostica dei modelli radiomici o sulle metodologie di estrazione delle feature, questo lavoro pone l'attenzione sugli aspetti computazionali dell'esecuzione delle pipeline. In particolare, l'obiettivo è verificare se applicazioni sviluppate originariamente per piattaforme x86 possano essere eseguite correttamente su architetture RISC-V senza modifiche sostanziali e quali interventi siano eventualmente necessari per garantirne il funzionamento.

Le due pipeline analizzate fanno uso di diverse librerie largamente utilizzate nel calcolo scientifico e nell'elaborazione di immagini mediche, tra cui NumPy, SimpleITK e PyRadiomics. Il processo di porting ha quindi rappresentato un'opportunità per valutare il livello di maturità di tali strumenti nell'ecosistema RISC-V e per identificare eventuali problematiche di

compatibilità che possono emergere durante l'esecuzione di applicazioni reali.

Un ulteriore obiettivo consiste nell'analizzare il comportamento delle pipeline al variare del livello di parallelismo. La piattaforma RISC-V utilizzata dispone infatti di 64 core logici, caratteristica che consente di studiare la scalabilità delle applicazioni e di valutare i benefici ottenibili attraverso l'esecuzione concorrente di più processi. Attraverso una serie di benchmark è stato quindi possibile misurare l'impatto del numero di worker sui tempi di esecuzione e sull'utilizzo delle risorse hardware disponibili.

Infine, il lavoro si propone di confrontare il comportamento delle due pipeline su RISC-V, evidenziando come differenti scelte implementative possano influenzare le prestazioni complessive. Sebbene entrambe siano basate su PyRadiomics e perseguano lo stesso obiettivo di estrazione delle feature radiomiche, esse adottano approcci differenti nella gestione dei dati e nelle operazioni di preprocessing, rendendole un interessante caso di studio per l'analisi prestazionale.

4.2 Problematiche di compatibilità e adattamenti per RISC-V

Sebbene le due pipeline analizzate siano state originariamente sviluppate in ambiente Python e facciano uso di librerie generalmente considerate multi-piattaforma, il loro porting su architettura RISC-V ha richiesto diversi interventi correttivi. Durante le prime esecuzioni sono infatti emersi numerosi errori di tipo *Illegal Instruction*, causati dall'esecuzione di istruzioni non correttamente supportate dalla piattaforma utilizzata o generate automaticamente dalle librerie scientifiche durante la compilazione e l'esecuzione.

Tali problematiche sono state osservate principalmente nelle librerie NumPy, SimpleITK e PyRadiomics, che costituiscono i componenti fondamentali delle pipeline radiomiche considerate in questo lavoro. Sebbene queste librerie siano ampiamente utilizzate su sistemi x86, il supporto all'architettura RISC-V risulta ancora meno maturo e può presentare incompatibilità dovute

te alla presenza di ottimizzazioni specifiche per determinate implementazioni hardware.

4.2.1 Compilazione conservativa di NumPy

La prima criticità ha riguardato NumPy, libreria ampiamente utilizzata per la gestione di array multidimensionali e operazioni numeriche vettorializzate. Le versioni precompilate disponibili tramite i normali canali di distribuzione causavano errori di esecuzione durante alcune operazioni interne utilizzate dalle pipeline.

Per risolvere il problema è stato necessario ricompilare NumPy direttamente dai sorgenti utilizzando opzioni di compilazione conservative, disabilitando le ottimizzazioni aggressive e la generazione automatica di codice vettoriale:

```
export CFLAGS="-O2 -march=rv64imafdc -mabi=lp64d -fno-tree-vectorize"
```

La scelta di utilizzare il flag `-fno-tree-vectorize` è legata alla necessità di impedire al compilatore di trasformare automaticamente alcune operazioni in istruzioni vettoriali che potevano generare incompatibilità durante l'esecuzione.

4.2.2 Disabilitazione dell'estensione vettoriale RVV

La piattaforma utilizzata implementa l'estensione vettoriale RVV (RISC-V Vector Extension), progettata per accelerare operazioni numeriche su grandi quantità di dati. Dal punto di vista teorico, tale estensione dovrebbe rappresentare un vantaggio per applicazioni scientifiche come quelle radio-miche, caratterizzate dall'elaborazione di immagini tridimensionali e matrici numeriche di grandi dimensioni.

Tuttavia, durante i test è stato osservato che alcune componenti di NumPy tentavano di utilizzare automaticamente istruzioni RVV, causando errori

di tipo *Illegal Instruction*. Per garantire la stabilità dell'esecuzione è stato quindi necessario disabilitare esplicitamente il supporto RVV mediante la variabile d'ambiente:

```
export NPY_DISABLE_CPU_FEATURES="RVV"
```

Un'indagine approfondita ha rivelato un'incompatibilità tra il compilatore e l'hardware in uso. Il compilatore di sistema (GCC) è configurato per generare codice conforme allo standard ufficiale ratificato RVV 1.0. Di contro, il processore implementa l'estensione vettoriale basata sulla versione preliminare draft RVV 0.7.1. La divergenza nei codici operativi delle istruzioni tra la versione 1.0 (software) e la versione 0.7.1 (hardware) impedisce la decodifica delle istruzioni da parte della CPU, rendendo necessaria la disattivazione dei registri vettoriali per garantire la stabilità dell'esecuzione.

4.2.3 Correzione di un'incompatibilità in PyRadiomics

L'ultima problematica ha riguardato la libreria PyRadiomics, in particolare durante la fase di validazione delle etichette all'interno delle maschere di segmentazione (nel modulo `imageoperations.py`). Nel codice originale della libreria, la verifica di validità dell'etichetta viene eseguita tramite il controllo di appartenenza:

```
if label not in labels:
```

Sulla piattaforma RISC-V, tuttavia, questa istruzione generava errori di esecuzione o comportamenti imprevisti. L'origine del problema risiede nella differente gestione della promozione automatica dei tipi e del casting dei tipi interi tra NumPy e il runtime Python su architettura RISC-V rispetto a x86. Nello specifico, mentre su x86 il confronto tra un intero nativo Python e gli interi specifici di NumPy (come `numpy.int64` o `numpy.int32`) viene gestito e risolto in modo trasparente, su RISC-V la discrepanza nella rappresentazione interna e nell'allineamento in memoria dei tipi a 32 e 64 bit impediva la

corretta valutazione di uguaglianza da parte dell'operatore di appartenenza `in`.

Per superare questa incompatibilità e garantire l'omogeneità dei tipi durante il confronto, è stato necessario forzare una conversione esplicita a tipi nativi Python. L'array di NumPy è stato convertito in una lista standard tramite il metodo `.tolist()` e i singoli elementi, insieme alla variabile `label`, sono stati trasformati in intero nativo:

```
labels_list = [int(x) for x in labels.tolist()]
if int(label) not in labels_list:
```

Sebbene la modifica sia localizzata e concettualmente semplice, l'introduzione di questa conversione esplicita ha risolto l'incompatibilità logica a basso livello, consentendo il corretto completamento del processo di estrazione delle feature sull'intera coorte di pazienti.

Nel complesso, gli interventi descritti hanno reso possibile l'esecuzione stabile delle pipeline radiomiche sulla piattaforma RISC-V utilizzata per gli esperimenti. Le problematiche incontrate suggeriscono che, nonostante la crescente diffusione dell'architettura RISC-V, alcune librerie scientifiche largamente utilizzate risultano ancora ottimizzate principalmente per ecosistemi più consolidati come x86. Il processo di adattamento effettuato in questo lavoro rappresenta quindi un caso di studio utile per valutare il livello di maturità dell'attuale ecosistema software RISC-V nel contesto dell'analisi di immagini mediche. Inoltre, le modifiche apportate alle librerie e agli strumenti utilizzati possono costituire un punto di partenza per il porting di altri codici basati sulle medesime dipendenze, favorendo future attività di migrazione verso piattaforme RISC-V di pipeline per l'analisi di immagini mediche.

Capitolo 5

Parallelizzazione delle pipeline di estrazione delle feature

5.1 Motivazioni della parallelizzazione

Uno degli obiettivi principali di questo lavoro è stato valutare le prestazioni di pipeline radiomiche reali su architettura RISC-V. Durante le prime esecuzioni dei software analizzati è emerso come l'elaborazione di grandi quantità di immagini mediche richiedesse tempi di esecuzione significativi, soprattutto a causa dell'elevato numero di feature radiomiche da calcolare per ciascun caso clinico.

La piattaforma hardware RISC-V utilizzata dispone tuttavia di un elevato numero di core di elaborazione. In un contesto di questo tipo, l'esecuzione sequenziale delle pipeline non consente di sfruttare appieno le risorse computazionali disponibili, lasciando inutilizzata gran parte della capacità di calcolo del sistema.

Per questo motivo è stata introdotta una strategia di parallelizzazione finalizzata a distribuire il carico di lavoro tra più processi concorrenti. L'obiettivo era duplice: ridurre il tempo complessivo di elaborazione dei dataset; valutare la capacità dell'architettura RISC-V di gestire carichi di lavoro parallelizzabili tipici delle applicazioni scientifiche.

L'approccio risulta particolarmente adatto al contesto radiomico poiché l'estrazione delle feature da immagini differenti costituisce un problema naturalmente parallelizzabile: l'elaborazione di un paziente non dipende infatti dai risultati ottenuti sugli altri pazienti, consentendo di eseguire più elaborazioni contemporaneamente senza introdurre dipendenze tra i processi.

5.2 Strategia di parallelizzazione adottata

L'analisi delle pipeline ha evidenziato che la parte più onerosa dal punto di vista computazionale è rappresentata dall'estrazione delle feature radiomiche. Ogni immagine viene elaborata in maniera indipendente e produce un insieme di misure numeriche che vengono successivamente raccolte in un file finale.

Sfruttando questa caratteristica, la parallelizzazione è stata realizzata secondo il paradigma task parallelism, assegnando a ciascun processo un sottoinsieme delle immagini da elaborare.

Invece di processare sequenzialmente tutti i casi clinici, il dataset viene suddiviso tra più worker, ciascuno dei quali esegue autonomamente l'intera procedura di estrazione delle feature sul proprio insieme di dati. Al termine dell'elaborazione, i risultati prodotti dai vari processi vengono aggregati per ottenere un unico output finale.

Questo approccio presenta diversi vantaggi: minimizza le modifiche necessarie al codice originale; evita la condivisione di strutture dati complesse tra processi; riduce il rischio di race conditions; consente una buona scalabilità all'aumentare del numero di core disponibili.

5.3 Implementazione

La parallelizzazione è stata implementata utilizzando il modulo multiprocessing di Python, che permette di creare processi indipendenti sfruttando effettivamente tutti i core della macchina. Per ciascuna esecuzione viene definito un numero configurabile di worker tramite il parametro:

```
--num_workers N
```

Il dataset viene quindi suddiviso tra i worker disponibili e ogni processo esegue in maniera indipendente l'estrazione delle feature radiomiche.

Lo schema operativo può essere riassunto come segue: caricamento della lista dei casi clinici; suddivisione dei casi tra i worker; avvio dei processi paralleli; estrazione delle feature da parte di ogni worker; raccolta dei risultati; generazione dell'output finale.

5.4 Considerazioni sull'architettura RISC-V

La parallelizzazione assume particolare rilevanza nel contesto della piattaforma RISC-V utilizzata. Sebbene le prestazioni del singolo core possano risultare inferiori rispetto a quelle di processori x86 di fascia alta, la disponibilità di numerosi core rende possibile compensare tale differenza attraverso l'esecuzione concorrente di più attività.

In applicazioni come la radiomica, caratterizzate da un elevato grado di parallelismo intrinseco, l'aumento del numero di processi permette di sfruttare più efficacemente le risorse hardware disponibili e di incrementare il throughput complessivo del sistema.

L'adozione di una versione parallelizzata delle pipeline ha quindi consentito non solo di ridurre i tempi di elaborazione, ma anche di valutare il comportamento dell'architettura RISC-V in scenari di calcolo scientifico ad alta intensità computazionale, evidenziandone la capacità di gestire carichi di lavoro distribuiti su più core.

Capitolo 6

Valutazione comparativa delle pipeline radiomiche

In questo capitolo vengono presentate e discusse le analisi sperimentali condotte sulle due pipeline radiomiche. I risultati vengono strutturati in due macro-sezioni: un'analisi statistica di accordo fra le feature estratte dalle due differenti implementazioni e una valutazione comparativa delle prestazioni computazionali (seriali e parallele) su architetture hardware x86 e RISC-V.

6.1 Confronto di riproducibilità delle feature radiomiche

In questa sezione viene presentata un'analisi rigorosa della riproducibilità e dell'accordo assoluto delle feature estratte dalle due pipeline di estrazione, estesa anche al confronto tra le implementazioni eseguite su architettura x86 e su architettura RISC-V. La valutazione è stata quantificata mediante il calcolo del Coefficiente di Correlazione Intraclassa (ICC) e del Coefficiente di Variazione (CV) a livello di singolo soggetto.

L'ICC [9] è un indice statistico ampiamente consolidato per la valutazione dell'affidabilità delle misure ripetute. Nello specifico, in questo studio è stato utilizzato il modello ICC(A, 1) (conosciuto anche come ICC2), strutturato su

un modello a due vie a effetti casuali (two-way random effects) per stimare l'accordo assoluto sulla singola misura. L'accordo assoluto dell'ICC penalizza attivamente qualsiasi scostamento numerico assoluto fra le due estrazioni e rappresenta quindi una scelta particolarmente appropriata per valutare la riproducibilità delle feature radiomiche ottenute da pipeline differenti. I valori di ICC sono compresi fra 0 e 1, dove valori superiori a 0.90 indicano un'affidabilità eccellente, mentre valori inferiori a 0.50 denotano un accordo scarso o insufficiente.

Il Coefficiente di Variazione viene utilizzato come misura complementare per esprimere la discrepanza relativa a livello di singolo soggetto (paziente_ROI). Definito come il rapporto fra la deviazione standard campionaria e la media aritmetica delle due estrazioni, il CV fornisce una quantificazione intuitiva dell'errore relativo percentuale. Ad esempio, un CV prossimo allo 0% indica un perfetto accordo assoluto.

6.1.1 Analisi di riproducibilità fra le pipeline

La discussione dei risultati ottenuti si avvale delle visualizzazioni grafiche e delle tabelle riassuntive riportate di seguito, volte a mappare le performance di accordo delle due pipeline, entrambe eseguite sulla medesima architettura, sia a livello di singola caratteristica sia a livello di intere classi di feature.

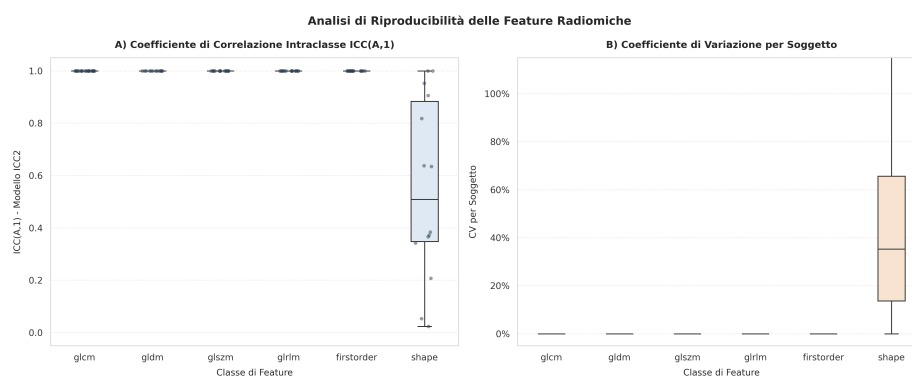


Figura 6.1: Distribuzione dei coefficienti ICC e CV per classe di feature.

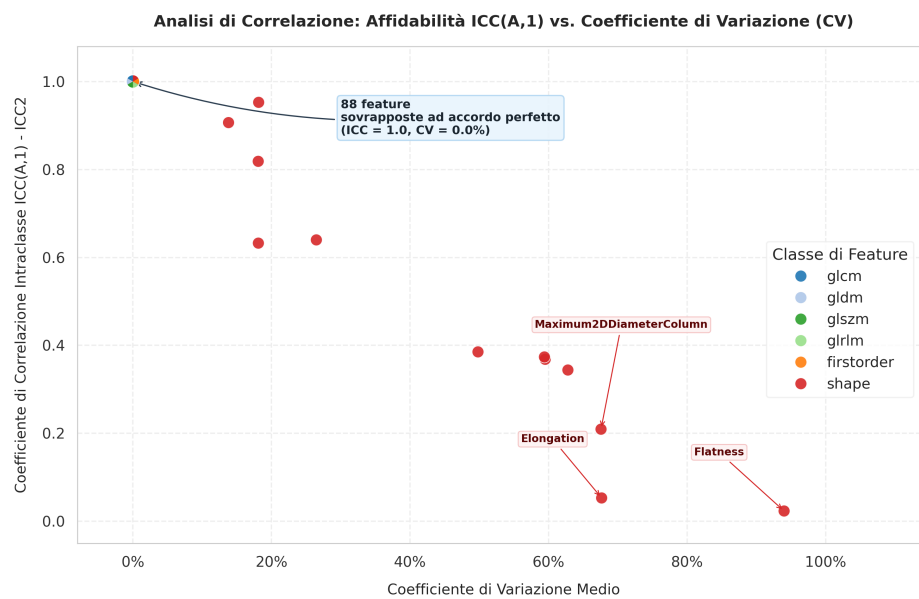


Figura 6.2: Grafico a dispersione della correlazione tra l'indice di affidabilità $ICC(A,1)$ e il Coefficiente di Variazione medio.

Feature di Forma	ICC(A,1)	CV Medio (%)	Affidabilità
Flatness	0.0234	94.07%	Scarsa
Elongation	0.0533	67.76%	Scarsa
Maximum2DDiameterColumn	0.2079	67.72%	Scarsa
Maximum2DDiameterRow	0.3421	62.86%	Scarsa
MajorAxisLength	0.3664	59.58%	Scarsa
Maximum3DDiameter	0.3720	59.46%	Scarsa
LeastAxisLength	0.3839	49.84%	Scarsa
Sphericity	0.6347	18.07%	Moderata
Maximum2DDiameterSlice	0.6379	26.58%	Moderata
SurfaceVolumeRatio	0.8180	18.05%	Buona
MinorAxisLength	0.9058	13.83%	Eccellente
SurfaceArea	0.9525	18.12%	Eccellente
MeshVolume	0.9999	0.15%	Eccellente
VoxelVolume	1.0000	0.00%	Eccellente

Tabella 6.1: Analisi di affidabilità (ICC) e accordo relativo (CV) per le feature morfologiche (classe Shape).

L'analisi della concordanza tra le feature radiomiche estratte dai due codici evidenzia un comportamento differente tra le varie classi di descrittori.

Come evidenziato nel grafico a dispersione (Figura 6.2), la quasi totalità delle feature estratte (88 su 98, pari al 90% del totale) si concentra in corrispondenza di un accordo perfetto ($ICC = 1.0000$ e $CV = 0.0\%$). Questo gruppo comprende l'intera gamma delle feature di tessitura (classi GLCM, GLRLM, GLSZM e GLDM), le feature statistiche di primo ordine e alcune di shape (VoxelVolume e MeshVolume). Questo risultato suggerisce che entrambe le pipeline elaborano in modo equivalente i valori di intensità contenuti all'interno della regione di interesse (ROI), producendo misure sostanzialmente sovrapponibili. Tali feature dipendono infatti principalmente dalla distribuzione delle intensità dei voxel e dalle relazioni statistiche locali tra essi, risultando quindi poco sensibili a eventuali differenze nella rappresentazione

geometrica o nell'orientamento spaziale dell'immagine.

Un comportamento differente emerge invece per alcune feature appartenenti alla categoria Shape, come mostrato nel grafico di Figura 6.1. Nella Tabella 6.1 le feature di forma mostrano livelli di accordo estremamente eterogenei, che possono essere compresi analizzando l'invarianza rotazionale degli algoritmi sottostanti. Una parte significativa delle feature di forma è basata sull'analisi degli assi principali della ROI, il cui calcolo matematico deriva da un'analisi di tipo PCA (Principal Component Analysis) applicata alle coordinate dei voxel della ROI. Feature quali Flatness ($ICC = 0.0234$, $CV = 94.07\%$), Elongation ($ICC = 0.0533$, $CV = 67.76\%$) e le lunghezze degli assi principali (Major, Minor e Least Axis Length) dipendono direttamente dal sistema di riferimento adottato e dall'orientamento dell'oggetto nello spazio.

La forte discrepanza osservata per queste specifiche feature può essere ricondotta a una differenza nella gestione dell'informazione spaziale tra le due pipeline. Nella seconda pipeline, l'orientamento delle immagini viene corretto mediante operazioni manuali di rotazione e riflessione dell'array successivamente alla conversione in formato NIfTI. Al contrario, nella prima pipeline, PyRadiomics utilizza direttamente i metadati geometrici associati all'immagine, quali orientamento e sistema di coordinate fisiche. Sebbene la segmentazione e la geometria globale della ROI rimangano sostanzialmente invariate, tali differenze nella rappresentazione spaziale possono influenzare il calcolo delle feature di shape basate sugli assi principali, particolarmente sensibili alla definizione del sistema di riferimento utilizzato.

È particolarmente interessante osservare che tutte le altre feature di forma, quali VoxelVolume, MeshVolume e Surface Area, mantengono invece valori di ICC molto elevati e coefficienti di variazione contenuti. Essendo proprietà geometriche assolute della ROI, il loro valore numerico rimane stabile anche in seguito alla rotazione spaziale eseguita nel secondo codice. Questo comportamento conferma che la discrepanza rilevata sulla classe Shape non è imputabile a un errore di calcolo del volume o della superficie della ROI,

bensì esclusivamente alla diversa gestione del sistema di riferimento spaziale e dell'allineamento degli assi principali.

6.1.2 Analisi di riproducibilità fra le architetture

Al fine di validare la correttezza matematica della pipeline di estrazione delle feature radiomiche anche in ambiente RISC-V, è stata effettuata un'analisi di riproducibilità confrontando i risultati ottenuti su tale architettura con quelli della medesima pipeline eseguita sulla piattaforma di riferimento x86.

I risultati mostrano una concordanza pressoché perfetta tra le due implementazioni. L'indice di correlazione intraclassa (ICC) risulta pari a 1 per tutte le classi di feature, mentre il coefficiente di variazione medio è trascurabile.

Tale comportamento indica che le due esecuzioni producono, a tutti gli effetti pratici, gli stessi valori di feature radiomiche, senza alcuna differenza statisticamente rilevante.

In conclusione, l'analisi conferma che il porting della pipeline su architettura RISC-V non introduce alcun bias computazionale, garantendo la piena equivalenza dei risultati rispetto all'implementazione su x86 e validando quindi la correttezza dell'intero processo di estrazione.

6.2 Confronto delle prestazioni computazionali

In questa sezione vengono analizzati i profili di consumo delle risorse computazionali del sistema, confrontando i tempi ottenuti mediante l'esecuzione dei test su architettura tradizionale x86 e sull'architettura emergente RISC-V, valutando i benefici in termini di scalabilità introdotti dalla parallelizzazione del codice.

6.2.1 Senza parallelizzazione

La Tabella 6.2 riporta i tempi ottenuti dall'esecuzione sequenziale delle due pipeline sulle architetture x86 e RISC-V.

Tempo di esecuzione (s)	Architettura x86		Architettura RISC-V	
	Pipeline 1	Pipeline 2	Pipeline 1	Pipeline 2
Totale	203.96	403.85	980.80	4097.21
Medio per Paziente/ROI	1.31	2.55	6.29	26.04

Tabella 6.2: Confronto delle metriche prestazionali tra le esecuzioni sequenziali delle pipeline su architetture x86 e RISC-V.

Per entrambe le pipeline si osserva innanzitutto un divario nei tempi di esecuzione tra le due architetture. Sulla piattaforma x86, la Pipeline 1 completa l'elaborazione in circa 204 s, mentre la Pipeline 2 richiede circa 404 s. Sulla piattaforma RISC-V gli stessi tempi aumentano rispettivamente a circa 981 s e 4097 s, evidenziando un rallentamento di circa 4,8 volte per la Pipeline 1 e di circa 10 volte per la Pipeline 2 rispetto alla piattaforma x86. Questo risultato è coerente con le differenze hardware tra le due architetture. La piattaforma x86 impiegata dispone infatti di frequenze operative più elevate, maggiore maturità del compilatore e di numerose ottimizzazioni software, mentre la piattaforma RISC-V utilizzata rappresenta una soluzione orientata principalmente alla sperimentazione e alla ricerca, con prestazioni inferiori ma elevata flessibilità.

Indipendentemente dall'architettura utilizzata, la Pipeline 1 risulta sistematicamente più veloce della Pipeline 2. Il tempo medio per paziente/ROI passa infatti:

- da 1.31 s a 2.55 s su x86;
- da 6.29 s a 26.04 s su RISC-V.

La Pipeline 2 richiede quindi circa il doppio del tempo necessario alla Pipeline 1 su architettura x86, e circa il quadruplo su architettura RISC-V. Tale

comportamento è prevedibile poiché la seconda pipeline esegue operazioni aggiuntive durante il pre-processing e il flusso di I/O, aumentando il costo computazionale complessivo. In particolare:

- Pipeline 1: Passa i percorsi dei file immagine e maschera direttamente all'estrattore di PyRadiomics, che delega interamente la lettura a SimpleITK, una libreria compilata in C++ che carica l'immagine da disco in modo ottimizzato direttamente in strutture di memoria contigue.
- Pipeline 2: Carica i file NIfTI in memoria Python usando la libreria Nibabel, generando un array NumPy. Successivamente, esegue operazioni di rotazione e riflessione speculare sul piano spaziale in ambiente Python tramite NumPy. Infine, converte l'array modificato in un oggetto SimpleITK tramite `sitk.GetImageFromArray()`. Questo flusso introduce tre principali overhead:
 - Overhead di I/O e Parsing in Python: Nibabel decodifica l'immagine in oggetti Python, operazione meno efficiente del caricamento nativo C++ di SimpleITK.
 - Overhead di Manipolazione: Eseguire rotazioni su array 3D di grandi dimensioni consuma cicli CPU preziosi in Python.
 - Overhead di Copia e Frammentazione: La rotazione di NumPy genera un array non contiguo in memoria. Quando viene chiamato `GetImageFromArray()`, SimpleITK è costretto ad allocare un nuovo buffer contiguo in C++ e copiare i dati elemento per elemento, raddoppiando l'allocazione e il tempo di trasferimento tra il runtime di Python e quello di C++.

6.2.2 Con parallelizzazione

Al fine di valutare quantitativamente l'efficacia del processo di parallelizzazione e confrontare l'efficienza computazionale sulle diverse architetture hardware, sono stati eseguiti test di scalabilità incrementando il numero di worker paralleli da 1 a 16. Il numero massimo di worker è stato limitato a 16 poiché il dataset utilizzato comprende un numero ridotto di pazienti; l'impiego di un numero superiore di processi avrebbe comportato uno sbilanciamento del carico di lavoro tra i worker, rendendo meno significativa la valutazione della scalabilità.

I risultati dettagliati di questa analisi prestazionale, che mettono a confronto l'architettura x86 con quella RISC-V per entrambe le implementazioni, sono riassunti nella Tabella 6.3 e illustrati graficamente nella Figura 6.3.

Piattaforma e Pipeline	Numero di Processi				
	1	2	4	8	16
x86 - Pipeline 1	203.74	109.40	64.09	45.14	34.07
x86 - Pipeline 2	403.60	223.17	123.78	85.01	65.21
RISC-V - Pipeline 1	977.8	527.5	283.0	174.1	114.0
RISC-V - Pipeline 2	4117.5	2213.9	1150.7	690.6	449.8

Tabella 6.3: Confronto dei tempi totali di esecuzione (Wall-clock) tra architetture x86 e RISC-V per le Pipeline 1 e 2 al variare del numero di processi.

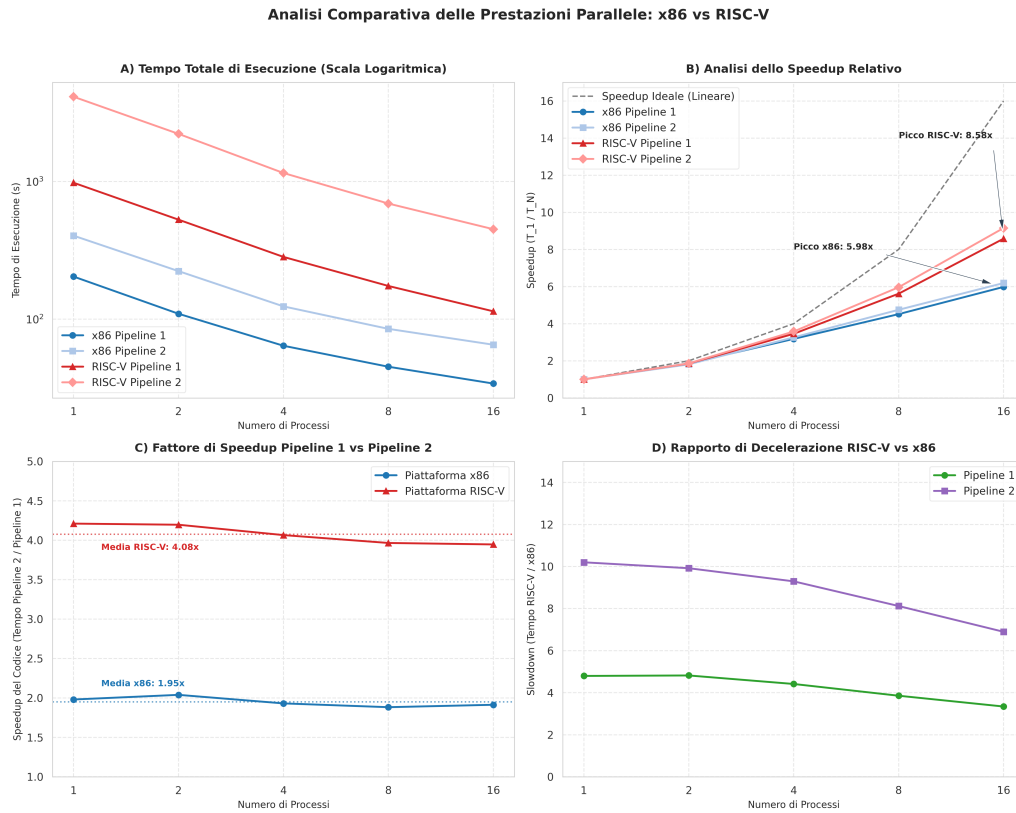


Figura 6.3: Analisi dettagliata delle performance di parallelizzazione confrontando le due piattaforme hardware (x86 e RISC-V) e le due implementazioni software (Pipeline 1 e Pipeline 2).

Il primo aspetto di interesse che emerge dall'analisi riguarda l'andamento dei tempi di esecuzione al crescere del livello di parallelizzazione: sulla piattaforma x86 il miglioramento rimane evidente fino a 16 core, riducendo il tempo totale a 34.07 s per la Pipeline 1 e a 65.21 s per la Pipeline 2, senza mostrare punti di inversione o saturazione precoce. Allo stesso modo, anche la piattaforma RISC-V continua a beneficiare in modo consistente dell'incremento delle risorse computazionali fino a 16 core, scendendo rispettivamente a 114.0 s per la Pipeline 1 e a 449.8 s per la Pipeline 2.

Questo aspetto è evidenziato anche dal grafico di Analisi di Speedup Relativo, che rivela una divergenza nel comportamento di scalabilità delle

due piattaforme. Su architettura x86, lo speedup di entrambe le pipeline cresce in modo continuo fino a 16 core, raggiungendo un fattore di circa $6\times$. Tuttavia, la curva mostra un andamento sub-lineare (lontano dallo speedup ideale) dovuto principalmente all'I/O su disco e all'overhead di spawn dei processi Python, che pesano proporzionalmente di più quando le singole tasks durano pochi secondi. La piattaforma RISC-V mostra invece una scalabilità decisamente più efficiente e vicina a quella lineare, superando gli $8.5\times$ di speedup per la Pipeline 1 e i $9.1\times$ per la Pipeline 2 a 16 core. Questo comportamento ottimale è favorito dai lunghi tempi di calcolo sequenziali su RISC-V, che rendono del tutto trascurabile l'overhead di parallelizzazione e massimizzano l'efficienza dei core della CPU.

Il terzo grafico illustra il rapporto tra i tempi di esecuzione delle due pipeline. Su entrambe le architetture la Pipeline 1 risulta costantemente più veloce della Pipeline 2. In particolare:

- su x86 la Pipeline 1 mantiene un vantaggio medio di circa $1.95\times$;
- su RISC-V il vantaggio medio sale a circa $4.08\times$.

Questo marcato divario nel comportamento del rapporto ($1.95\times$ su x86 contro $4.08\times$ su RISC-V) costituisce un punto di analisi fondamentale. Mentre l'architettura x86 riesce a mitigare l'overhead software introdotto dalla Pipeline 2 grazie a una suite di compilatori maturi e all'uso estensivo di istruzioni vettoriali (SIMD) per le operazioni su array di NumPy, la piattaforma RISC-V risente notevolmente di questi passaggi intermedi. La copia dei dati tra formati diversi e la manipolazione di array non ottimizzata a livello hardware su RISC-V degradano le prestazioni della Pipeline 2 in modo sproporzionato.

Tuttavia, il fatto che il rapporto rimanga pressoché costante al variare del numero di core indica che la parallelizzazione beneficia entrambe le implementazioni in maniera simile, mentre la differenza prestazionale deriva principalmente dalle diverse scelte implementative dei due algoritmi piuttosto che dal meccanismo di parallelizzazione. In altre parole, la parallelizzazio-

ne non modifica il rapporto di efficienza tra le due pipeline, ma ne riduce proporzionalmente il tempo di esecuzione.

Il quarto grafico confronta direttamente le due piattaforme. Con un solo core, l'architettura RISC-V richiede tempi circa:

- 4.8 volte superiori per la Pipeline 1;
- 10.2 volte superiori per la Pipeline 2.

Tale differenza è attribuibile principalmente alle differenti caratteristiche hardware delle due piattaforme, in particolare alla minore frequenza operativa e alle inferiori prestazioni per singolo core del sistema RISC-V rispetto al processore x86. Tuttavia, aumentando il numero di core, il divario si riduce progressivamente. A 16 core il rallentamento della piattaforma RISC-V scende a circa:

- $3.4\times$ per Pipeline 1;
- $6.9\times$ per Pipeline 2.

Nel caso della Pipeline 1 ciò corrisponde a una riduzione del divario prestazionale di circa il 30% rispetto all'esecuzione seriale, mentre per la Pipeline 2 il divario si riduce di circa il 32%, passando da $10.2\times$ a $6.9\times$. Questo risultato evidenzia come la parallelizzazione sia particolarmente efficace sulla piattaforma RISC-V, consentendo di compensare in parte le inferiori prestazioni del singolo core attraverso un utilizzo più efficiente delle risorse multi-core.

In conclusione, i risultati dello studio di fattibilità condotto in questo lavoro di tesi dimostrano che la strategia di parallelizzazione adottata risulta efficace su entrambe le architetture, consentendo una significativa riduzione dei tempi di esecuzione. La piattaforma RISC-V, pur presentando prestazioni assolute inferiori per singolo processo, dimostra una scalabilità parallela eccezionale al crescere dei processi impiegati, riducendo sensibilmente il proprio divario rispetto alla piattaforma x86. L'interesse verso le architetture

RISC-V non risiede quindi esclusivamente nelle prestazioni assolute di calcolo sequenziale, ma nella loro elevata scalabilità e nell'efficienza con cui riescono a sfruttare l'hardware multicore. Questa caratteristica le rende particolarmente adatte a classi di applicazioni intrinsecamente parallelizzabili, come l'estrazione di feature radiomiche, soprattutto in scenari in cui risultano rilevanti la natura aperta e personalizzabile dell'ISA e il potenziale in termini di efficienza energetica delle soluzioni basate su architetture RISC-V.

Conclusioni

L'obiettivo di questa tesi era dimostrare che il confronto tra differenti implementazioni di pipeline radiomiche non dovesse limitarsi alla sola correttezza delle feature estratte, ma comprendere anche aspetti di portabilità, efficienza computazionale e adattabilità a nuove architetture hardware. Attraverso il porting di due pipeline radiomiche su architettura RISC-V, l'analisi della concordanza delle feature prodotte e il confronto prestazionale con una piattaforma x86, è stato possibile valutare congiuntamente la correttezza funzionale delle implementazioni e il loro comportamento dal punto di vista computazionale.

Il primo risultato significativo del lavoro riguarda la fattibilità del porting. Sebbene le pipeline fossero state sviluppate originariamente per sistemi x86, è stato possibile adattare all'architettura RISC-V intervenendo sulle principali criticità emerse durante l'esecuzione. In particolare, le modifiche apportate alle librerie scientifiche e alla configurazione dell'ambiente software hanno consentito di ottenere un'esecuzione stabile e riproducibile, evidenziando al tempo stesso come l'ecosistema RISC-V, pur in continua evoluzione, presenti ancora alcuni limiti in termini di maturità e compatibilità.

L'analisi della concordanza tra le feature estratte dalle due implementazioni ha inoltre mostrato come, nonostante le differenti scelte implementative adottate nelle fasi di caricamento, preprocessing e gestione dei dati, la quasi totalità delle feature radiomiche prodotte risulti sostanzialmente equivalente. Le differenze osservate hanno interessato esclusivamente alcune feature morfologiche sensibili all'orientamento spaziale della regione di interesse,

confermando che gli scostamenti rilevati derivano dalle differenti modalità di elaborazione delle immagini e non da errori nel processo di estrazione. Tale risultato costituisce un'importante verifica della correttezza delle due pipeline e dimostra come implementazioni differenti possano produrre risultati quantitativi comparabili.

Dal punto di vista prestazionale, il confronto tra le due pipeline ha evidenziato come le scelte implementative influenzino significativamente i costi computazionali. La pipeline caratterizzata da un flusso di elaborazione più diretto ha mostrato tempi di esecuzione sistematicamente inferiori rispetto all'implementazione che introduce ulteriori operazioni di preprocessing e manipolazione dei dati. Questo risultato conferma come l'organizzazione interna di una pipeline radiomica rappresenti un elemento determinante per l'efficienza complessiva del sistema.

Il confronto tra le piattaforme hardware ha mostrato che l'architettura RISC-V presenta prestazioni inferiori rispetto alla piattaforma x86 utilizzata come riferimento, evidenziando un divario riconducibile sia alle differenti caratteristiche hardware sia al diverso grado di maturità dell'ecosistema software. Tuttavia, gli esperimenti hanno dimostrato che tale architettura è in grado di eseguire correttamente applicazioni radiomiche reali e di beneficiare significativamente dell'introduzione del parallelismo, sfruttando efficacemente la disponibilità di numerosi core di elaborazione. I risultati ottenuti suggeriscono quindi che, pur non rappresentando ancora un'alternativa prestazionale alle piattaforme x86 più mature, RISC-V costituisca una soluzione promettente per applicazioni scientifiche che possano trarre vantaggio dalla sua natura aperta e dalla sua elevata scalabilità.

Tra gli elementi di maggiore interesse del presente lavoro vi è l'approccio integrato adottato per la valutazione delle pipeline. A differenza di studi focalizzati esclusivamente sull'accuratezza delle feature radiomiche o sulle prestazioni hardware, questa tesi affronta congiuntamente la correttezza dei risultati, la portabilità delle implementazioni e l'analisi prestazionale, fornendo una valutazione complessiva del comportamento di due pipeline radiomi-

che su differenti architetture. Tale approccio consente di evidenziare non solo l'affidabilità delle implementazioni considerate, ma anche l'impatto concreto delle scelte progettuali sulle prestazioni del sistema.

Il lavoro presenta tuttavia alcune limitazioni. Le analisi sono state condotte su un numero limitato di pipeline e su un unico dataset di immagini di risonanza magnetica mammaria; pertanto, l'estensione dei risultati ad altri contesti applicativi richiede ulteriori verifiche sperimentali. Inoltre, le prestazioni osservate su RISC-V sono inevitabilmente influenzate dall'attuale stato di sviluppo delle librerie scientifiche disponibili per questa architettura, che in molti casi risultano ancora meno ottimizzate rispetto alle corrispondenti implementazioni per x86.

Da queste considerazioni derivano diverse possibili linee di sviluppo futuro. Un primo ambito di interesse riguarda l'estensione dello studio ad altre pipeline radiomiche e a differenti modalità di imaging medico, al fine di verificare la generalizzabilità dei risultati ottenuti. Un secondo sviluppo consiste nell'approfondimento delle ottimizzazioni specifiche per RISC-V, valutando il potenziale delle estensioni vettoriali e delle future evoluzioni dell'ecosistema software. Infine, ulteriori attività potranno essere dedicate all'ottimizzazione delle pipeline stesse, riducendo gli overhead introdotti dalle operazioni di preprocessing e migliorando ulteriormente la scalabilità delle versioni parallelizzate. In questo lavoro l'attenzione è stata rivolta principalmente all'aumento del throughput, ossia alla massimizzazione del numero di pazienti elaborati nell'unità di tempo attraverso la parallelizzazione a livello di paziente. Un'interessante direzione di sviluppo consiste invece nell'esplorare un approccio complementare basato sulla parallelizzazione delle elaborazioni interne al singolo paziente, ad esempio distribuendo il processamento delle diverse ROI o l'estrazione delle singole feature. Le due strategie rispondono tuttavia a esigenze differenti e comportano un compromesso tra massimizzazione del throughput e minimizzazione della latenza.

Nel complesso, il lavoro svolto dimostra che il porting di pipeline radiomiche su architettura RISC-V è tecnicamente realizzabile e che, una volta

affrontate le principali criticità di compatibilità, è possibile ottenere un'esecuzione stabile mantenendo la sostanziale equivalenza delle feature estratte. I risultati ottenuti rappresentano un contributo alla valutazione dell'impiego di RISC-V nel calcolo scientifico e mostrano come questa architettura possa costituire una piattaforma promettente per future applicazioni nel campo dell'elaborazione di immagini mediche e della radiomica.

Bibliografia

- [1] PyRadiomics Documentation: <https://pyradiomics.readthedocs.io/>.
- [2] van Griethuysen, J. J. M., Fedorov, A., Parmar, C., et al. (2017). Computational Radiomics System to Decode the Radiographic Phenotype. *Cancer Research*, 77(21), e104–e107. <https://doi.org/10.1158/0008-5472.CAN-17-0339>.
- [3] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. *Nature* 585, 357–362 (2020). <https://doi.org/10.1038/s41586-020-2649-2>
- [4] R. Beare, B. C. Lowekamp, Z. Yaniv, “Image Segmentation, Registration and Characterization in R with SimpleITK”, *J Stat Software*, 86(8), <https://doi.org/10.18637/jss.v086.i08>, 2018.
- [5] Gregory R. Lee, Ralf Gommers, Filip Wasilewski, Kai Wohlfahrt, Aaron O’Leary (2019). PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software*, 4(36), 1237, <https://doi.org/10.21105/joss.01237>.
- [6] Brett, M., Markiewicz, C. J., Hanke, M., Côté, M.-A., Cipollini, B., Papadopoulos Orfanos, D., McCarthy, P., Jarecka, D., Cheng, C. P., Larson, E., Halchenko, Y. O., Cottaar, M., Ghosh, S., Wassermann, D., Gerhard, S., Lee, G. R., Baratz, Z., Moloney, B., Wang, H.-T., ...

- freec84. (2025). nipy/nibabel: 5.3.3 (5.3.3). Zenodo. <https://doi.org/10.5281/zenodo.17833605>
- [7] Scapicchio C., Gabelloni M., Barucci A., et al. *A deep look into radiomics*. La Radiologia Medica, vol. 126, pp. 1296–1311, 2021. <https://doi.org/10.1007/s11547-021-01389-x>.
- [8] Morcaldi, Gianluca, et al. "Efficient machine learning models leveraging DCE-MRI morphological and dynamic features allow accurate breast lesion classification." *Biomedical Physics & Engineering Express* 12.2 (2026): 025061. <https://doi.org/10.1088/2057-1976/ae5712>.
- [9] Liljequist, D., Elfving, B., & Skavberg Roaldsen, K. (2019). Intra-class correlation - A discussion and demonstration of basic features. PLOS ONE, 14(7), e0219854. <https://doi.org/10.1371/journal.pone.0219854>.
- [10] Daniels, D., Last, D., Cohen, K., Mardor, Y., & Sklair-Levy, M. (2024). Standard and Delayed Contrast-Enhanced MRI of Malignant and Benign Breast Lesions with Histological and Clinical Supporting Data (Advanced-MRI-Breast-Lesions) (Version 2) [dataset]. The Cancer Imaging Archive. <https://doi.org/10.7937/C7X1-YN57>.

Ringraziamenti

Desidero ringraziare il mio relatore, il Professor Andrea Omicini, e i miei correlatori, Francesco Giacomini e Camilla Scapicchio, per il tempo che mi hanno dedicato, per la costante disponibilità e per il prezioso supporto offertomi durante tutto il percorso di questa tesi.