Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Informatica

Approcci di deep learning per l'interpolazione delle proiezioni nella ricostruzione tomografica

Relatrice:
Prof.ssa
Elena Morotti

Presentata da:
Francesco Tomba

Correlatrice:
Chiar.ma Prof.ssa
Elena Loli Piccolomini

II sessione
Anno Accademico 2024/2025

Abstract

Le analisi effettuate tramite tomografia computerizzata possono comportare numerosi rischi per i pazienti, a causa dell'elevato numero di proiezioni necessarie. Nonostante esistano tecniche per abbassare il quantitativo di proiezioni, esse presentano ancora numerosi difetti in termini di rumore e artefatti grafici. L'obiettivo di questa tesi consiste nella realizzazione, tramite reti neurali, di una tecnica di generazione artificiale di proiezioni, in grado di aumentare la quantità di proiezioni a disposizione, senza ulteriori rischi sul paziente. Si sono confrontati i risultati di due principali modelli, i quali differiscono per il numero di proiezioni in input. Sono quindi stati riportati i risultati ottenuti con metodologie naive note in letteratura, poi confrontati con i risultati acquisiti tramite reti neurali. Le metriche riportate mostrano un significativo miglioramento della qualità delle ricostruzioni, rispetto alle originali, raggiungendo un livello di dettaglio significativamente più vicino all'oggetto reale. Queste implementazioni mostrano un potenziale significativo per l'utilizzo di queste tecnologie nelle architetture tomografiche moderne.

Introduzione

Il deep learning e le reti neurali stanno rivoluzionando sempre di più l'approccio a problemi della comunità scientifica e della vita di tutti i giorni. Un campo di applicazione sempre più florido per queste tecnologie è l'elaborazione e la ricostruzione di immagini. In questa tesi si affronta l'applicazione di una rete convoluzionale al settore medico, esplorando metodi per il miglioramento della qualità delle immagini di Tomografia Computerizzata (in inglese *Computed Tomography*, CT). Fra gli aspetti più critici delle attuali scansioni CT troviamo sicuramente l'aspetto di esposizione ai raggi X, che, essendo ionizzanti, rappresentano un potenziale rischio per la salute all'aumentare dei livelli di esposizione. Un altro aspetto fondamentale da considerare è il tempo necessario alla raccolta dei dati, che se eccessivamente lungo può portare a movimenti involontari da parte del paziente, con conseguente peggioramento delle ricostruzioni. Per ridurre al minimo questi rischi, si stanno sempre più delineando protocolli alternativi alla CT classica, come ad esempio la tomografia a viste sparse (*Sparse-view Computed Tomography*), con la quale si raccolgono solo parte delle proiezioni necessarie.

D'altra parte, per ottenere maggiore qualità e scansioni migliori, è necessario aumentare la quantità di proiezioni raccolte, con conseguente aumento di esposizione ai raggi e di tempi di raccolta dei dati. L'obiettivo principale di questa tesi consiste nell'aumentare artificialmente la quantità delle proiezioni, utilizzando come base i dati realmente raccolti con la tomografia sparse-view, senza ulteriori rischi sul paziente. Questo obiettivo è perseguito attraverso la generazione di proiezioni sintetiche, ottenute mediante reti neurali e collocate

come proiezioni intermedie tra quelle originali. Rispetto ai metodi analitici di interpolazione classica, gli approcci basati sulle reti neurali rappresentano un'alternativa efficace, in grado di produrre proiezioni con dettagli più nitidi e con una migliore definizione strutturale. La tesi si concentra sugli aspetti computazionali e di ottimizzazione necessari per adattare architetture convoluzionali al dominio del problema, affrontando problematiche legate alla gestione della memoria e alla gestione del diverso numero di proiezioni considerate per ciascuna generazione sintetica. Nel Capitolo 1 vengono introdotti i concetti fondamentali della tomografia computerizzata, con un approfondimento sulle principali tecniche impiegate. Il capitolo prosegue con un'introduzione alle reti neurali e si conclude con l'analisi del caso di studio relativo all'interpolazione delle proiezioni. Il Capitolo 2 descrive la metodologia proposta, soffermandosi sugli aspetti tecnici riguardanti l'applicazione delle reti neurali alla tomografia 3D, la quale richiede accorgimenti specifici per la gestione dell'elevato volume di dati e dell'utilizzo della memoria. Nel Capitolo 3 viene illustrata la configurazione sperimentale, mentre nel Capitolo 4 sono riportati i risultati ottenuti, che dimostrano il potenziale delle soluzioni implementate nel miglioramento delle ricostruzioni tridimensionali.

Indice

Capitolo 1

Introduzione alla tomografia computerizzata e alle reti neurali

In questo capitolo si introducono i principi alla base della tomografia computerizzata e si presentano le principali architetture e componenti delle reti neurali convoluzionali.

1.1 La tomografia computerizzata

La tomografia computerizzata è una tecnica di imaging medico con la quale è possibile ricostruire una rappresentazione tridimensionale dell'oggetto della scansione [1]. Essa basa il suo funzionamento sull'utilizzo di raggi X che, attraversando il corpo, tendono a venir assorbiti ed attenuati, e quindi a venir ricevuti con intensità diverse da un sensore detto detector. Tramite queste variazioni, è possibile identificare l'attenuazione del materiale attraversato, e quindi dedurre la densità e la composizione del tessuto.

1.1.1 Fasi del processo CT

Alla base della CT sono presenti varie fasi di raccolta ed elaborazione dei dati, al fine di ricostruire un volume 3D adatto all'applicazione desiderata. Il concetto che rende possibile la ricostruzione di questi volumi è quello di proiezione, definita come la misura dell'attenuazione dei raggi X, durante l'attraversamento del corpo. L'immagine risultante quindi apparirà come una vista da un certo angolo del corpo analizzato, in scala di grigi. Un'area tendente al bianco indicherà una maggiore attenuazione del materiale attraversato, mentre un'area prossima al nero indicherà l'opposto.

Prelevando un cospicuo numero di proiezioni da diverse angolazioni, è possibile ricostruire il volume originale, attraverso l'applicazione di tecniche come FBP o FDK. È importante notare come la quantità di proiezioni incida direttamente sulla qualità del volume ricostruito, dove un numero troppo basso di proiezioni può portare ad artefatti o rumore eccessivo.

1.1.2 CT a ventaglio - fan beam

Durante l'evoluzione di questi sistemi, sono stati impiegati svariati approcci differenti per la collezione di tutte le proiezioni necessarie. Con l'avvento della terza generazione di CT, è stato introdotto l'utilizzo di un fascio di raggi X detto a forma di ventaglio (*fan beam*).

Come illustrato in Figura 1.1, in questa configurazione, la sorgente e il detector vengono ruotati insieme attorno al paziente, mentre il fascio di raggi si espande in molte direzioni partendo da un punto focale unico. Il detector è disposto di fronte alla sorgente in una forma ad arco.

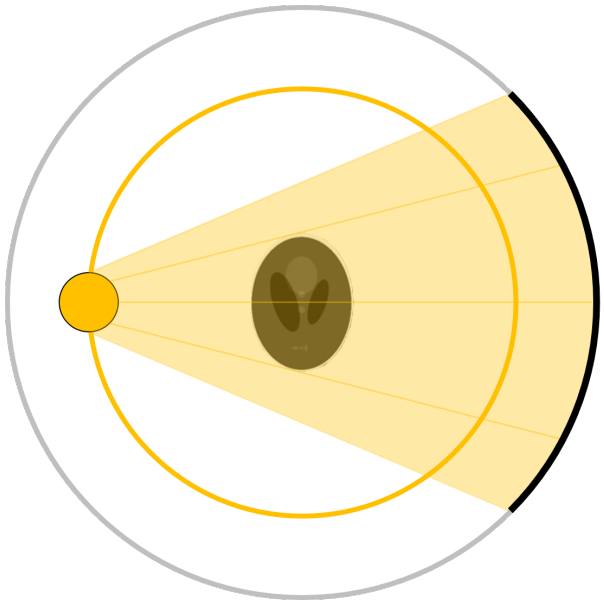


Figura 1.1: Acquisizione di una proiezione con metodologia fan beam

Fissata una sezione bidimensionale del volume (*slice*), questo metodo produce una proiezione di quella slice dall'angolazione selezionata. Ripetendo il processo da diverse angolazioni è possibile ricostruire una slice, e ripetendo il processo per tutte le slice del volume, è possibile ricostruirlo riga per riga. Per accelerare il processo sono state sviluppate le CT multistrato, che permettono di analizzare più righe contemporaneamente. Tra i vantaggi più significativi della tecnologia fan beam, risalta l'eccellente contrasto dei tessuti molli, che permette di distinguere più facilmente strutture di densità simile all'interno del corpo analizzato.

1.1.3 CT a forma conica - cone beam

Un'altra tipologia di tomografie largamente utilizzate sono le tomografie computerizzate a forma conica (*cone beam*, CBCT), in cui il fascio di raggi X utilizzato ha forma conica. Si riporta un esempio di utilizzo di questa metodologia nella Figura 1.2.

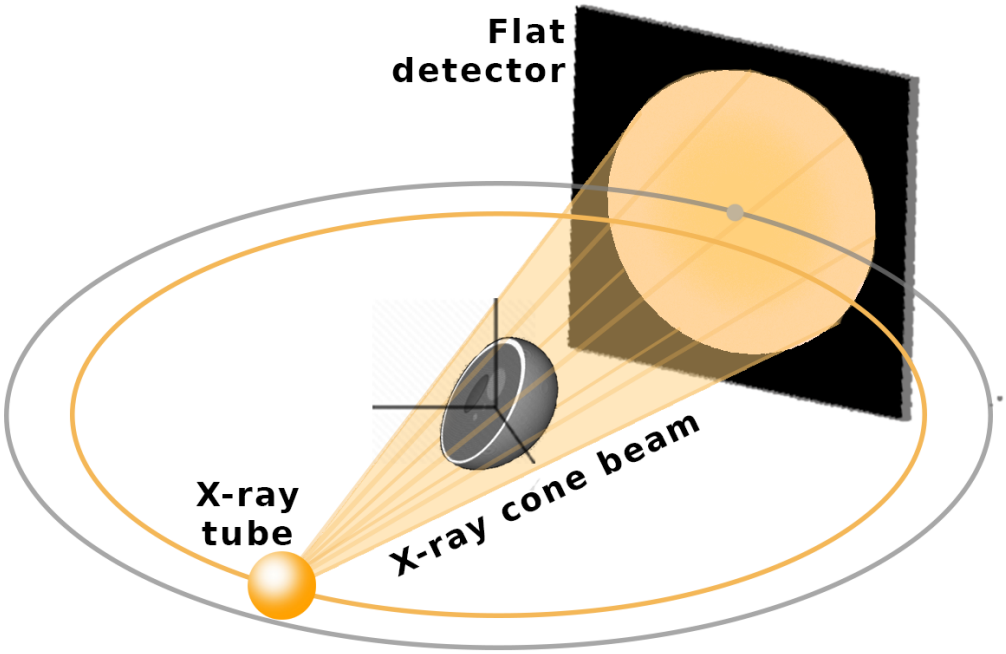


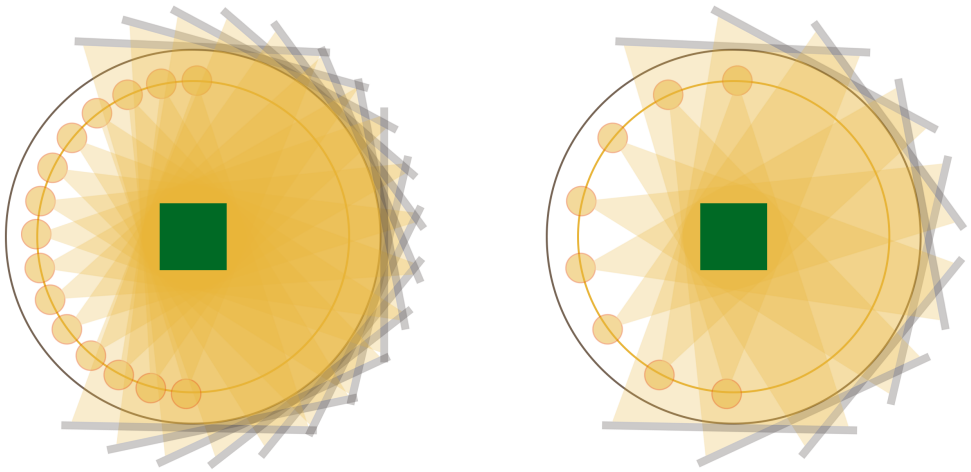
Figura 1.2: Acquisizione di una proiezione con metodologia CBCT

Questa configurazione permette di acquisire dati sull'intero volume con una singola applicazione, producendo in uscita una proiezione 2D che coinvolge tutti gli strati del volume originale. Di conseguenza, tramite questa tecnica è sufficiente una singola rotazione intorno al volume per riuscire a ricostruire tutto il corpo, riducendo drasticamente i tempi di esecuzione dell'analisi rispetto all'approccio fan beam. Questa metodologia, inoltre, consente una riduzione sostanziale della dose radiante, a fronte di un minor contrasto tra tessuti molli, limitandone quindi l'applicazione in contesti specifici. In questo caso il detector utilizzato è bidimensionale planare.

1.1.4 Tomografia a viste sparse - sparse view

Le scansioni CT presentano diverse criticità alla base del loro funzionamento. Tra gli aspetti più rilevanti troviamo l'esposizione ai raggi X, i quali, essendo radiazioni ionizzanti, comportano un rischio potenziale per la salute

che aumenta proporzionalmente alla dose somministrata [2]. Un'altra limitazione significativa riguarda il tempo necessario per l'acquisizione dei dati. Infatti le scansioni particolarmente prolungate possono indurre movimenti involontari del paziente, come respirazione o piccoli aggiustamenti posturali, che possono tradursi in artefatti e degradazione della qualità delle immagini ricostruite. Entrambi questi fattori sono da considerarsi parte del processo di scansione CT, ma esistono tecniche per mitigare questi rischi. A questo fine si adottano tecniche come la sparse-view CT, con la quale si raccolgono solo parte delle proiezioni normalmente necessarie, mantenendo comunque un intervallo regolare fra gli angoli.



(a) CT classica (b) CT sparse-view

Figura 1.3: Riduzione delle proiezioni tramite sparse-view

Tramite questa tecnica, illustrata in Figura 1.3, i rischi sopracitati si riducono ampiamente, tuttavia un quantitativo ridotto di proiezioni riduce la qualità della ricostruzione, introducendo rumore e artefatti all'interno della ricostruzione.

1.1.5 Ricostruzione dei volumi - FBP e FDK

Il passaggio fondamentale che permette la ricostruzione è la retroproiezione. Questo procedimento permette di ricostruire una slice del volume per volta, utilizzando tutte le proiezioni a disposizione per quella slice. Il metodo di applicazione consiste nel distribuire i valori presenti nella proiezione lungo il tragitto percorso dai raggi del fascio, in modo corrispondente alla geometria utilizzata.

Il risultato della semplice retroproiezione però presenta significative sfocature nei volumi ricostruiti. Per correggere questo effetto, si è introdotto il metodo FBP (*Filtered Back Projection* [3]). Questo metodo prevede l'applicazione di un filtro, tipicamente a rampa o simili varianti attenuate, su tutte le proiezioni in input. Lo scopo del filtraggio è esaltare le componenti ad alta frequenza nelle proiezioni, per aumentare la nitidezza dell'immagine finale. Solo dopo questa fase le proiezioni vengono retroproiettate, ottenendo una ricostruzione accurata della slice corrente. L'algoritmo FBP è matematicamente esatto per geometrie parallele o fan-beam.

Per geometrie tridimensionali cone-beam, è necessario un approccio differente. In questo contesto viene impiegato l'algoritmo FDK (*Feldkamp–Davis–Kress* [4]), una generalizzazione approssimata di FBP. Esso introduce correzioni geometriche che compensano la normale divergenza del fascio conico, applicando un filtraggio dedicato. A causa dell'introdotta approssimazione, a differenza del metodo FBP, il metodo FDK non risulta essere un'inversione esatta dalle proiezioni al volume. Per questo motivo, anche con infinite proiezioni, non sarebbe comunque possibile ottenere una ricostruzione perfetta del volume originale.

1.2 Reti neurali convoluzionali

Le reti neurali convoluzionali (CNN) sono tipologie di reti neurali utilizzate principalmente nel riconoscimento e nell'elaborazione di immagini. Il loro scopo principale è quello di identificare pattern frequenti e distintivi, as-

sociandoli al risultato atteso in output, cercando ed estrapolando connessioni profonde fra queste. Le CNN infatti ispirano il loro funzionamento alle strutture del sistema visivo animale, le quali analizzano le immagini attraverso un metodo gerarchico, inizialmente identificando caratteristiche di basso livello come bordi o angoli, per poi combinarle per identificarne di più complesse.

1.2.1 Componenti principali nelle CNN

Le reti neurali sono divise in componenti chiamati strati (o *layer*), che sono unità di trasformazione dei dati di input in dati di output secondo regole stabilite. Inoltre i layer possono essere parametrici o non parametrici. La differenza risiede nel fatto che i layer parametrici variano il proprio output in base ad alcuni valori numerici interni modificabili, mentre i non parametrici no. I layer di cui sono principalmente composte le CNN sono detti convolutivi, ovvero layer parametrici composti da uno o più filtri (detti *kernel*) che vengono ripetutamente applicati su tutta l'immagine, fino alla totale copertura di quest'ultima. Il metodo di applicazione di un kernel è la moltiplicazione tra esso e la porzione di figura selezionata. Questa operazione fornirà in uscita un valore unico, somma di tutti i valori moltiplicati. Il risultato di un layer convolutivo è detta *feature map*, ovvero un insieme di matrici che rappresentano la visione dell'immagine attraverso un kernel, dove quest'ultimo ha evidenziato particolari caratteristiche. Altri layer di fondamentale importanza sono detti di *pooling*, ovvero layer non parametrici specifici che si occupano di aggregare in un unico valore informazione proveniente da più punti dell'immagine. Tra i più utilizzati vediamo MaxPooling e AvgPooling, che rispettivamente restituiscono il valore massimo e la media dei valori trovati in una certa porzione dell'immagine. Questi sono principalmente utilizzati per ridurre la dimensione spaziale dell'informazione di input e quindi per ridurre la complessità della rete nel suo insieme.

1.2.2 Funzioni di attivazione

Sono anche utilizzate funzioni di attivazione, posizionate dopo ogni operazione lineare, come la convoluzione. Queste funzioni sono non lineari, quindi la loro applicazione garantisce non linearità fra operazioni lineari. Una delle funzioni di attivazione più utilizzate è la ReLU, ovvero una funzione che mappa a 0 i valori di input negativi e mantiene invariati i valori positivi.

1.2.3 Skip connections

Le connessioni residue, o *skip connections*, sono elementi di fondamentale importanza per alcune architetture moderne. Alcune di esse, come la UNet o ResNet basano il loro intero funzionamento su questa tecnica.

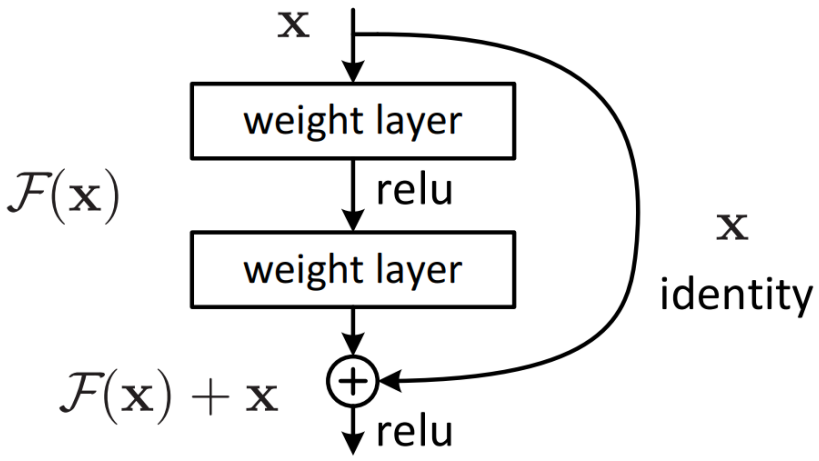


Figura 1.4: Schema di funzionamento delle skip connections

Come riportato nella Figura 1.4, essa consiste in una connessione diretta che attraversa più layer già presenti della rete, sommando l'input originale all'output di uno strato più profondo [5]. Questo approccio mitiga il problema della scomparsa del gradiente, migliorandone la propagazione ed accelerando l'allenamento delle reti. Un altro vantaggio di questo approccio risiede della preservazione dei dettagli spaziali, infatti tramite questa tecnica è possibi-

le recuperare informazioni che sarebbero altrimenti andate perse durante le convoluzioni o le operazioni di riduzione.

1.2.4 Layer di normalizzazione

Un'altra componente importante nelle CNN è rappresentata dai layer di normalizzazione, introdotti per migliorare la stabilità e l'efficienza dell'allenamento. A questo scopo questi layer regolarizzano le distribuzioni interne dei dati in modo che abbiano caratteristiche più uniformi, riducendo così potenziali variazioni di scala dei dati e accelerando la convergenza della rete.

Tra i metodi più noti vi è la **Batch Normalization** [6] (BatchNorm), che normalizza gli input di ciascun layer calcolando media e varianza all'interno di un mini-batch.

Un'alternativa alla BatchNorm è la **Group Normalization** [7] (GroupNorm), che non basa il suo funzionamento sul batch ma divide i canali delle feature map in gruppi, normalizzando ciascuno separatamente. Questo approccio risulta particolarmente efficace in scenari in cui la dimensione del batch è ridotta, mantenendo buone prestazioni anche in presenza di vincoli di memoria.

1.2.5 Allenamento del modello

Al momento della creazione in memoria del modello, i layer parametrici otterranno parametri interni casuali. Per questo motivo i risultati iniziali della rete appariranno casuali, o solo in parte coerenti. Durante un processo di miglioramento iterativo della rete, detto allenamento, si ricercano i valori migliori per ogni parametro. Questo processo viene guidato da una costante comparazione fra il risultato attuale della rete e il risultato atteso, seguito da una modifica dei parametri basata sugli errori appena identificati. Le funzioni di comparazione sono dette funzioni di perdita (*loss functions*).

1.2.6 L'architettura UNet

L'architettura UNet è un modello di rete neurale estremamente consolidato e studiato negli ultimi anni. Essa è stata introdotta da Olaf Ronneberger, Philipp Fischer e Thomas Brox [8], e il suo scopo principale è eseguire segmentazione di immagini, ovvero individuazione e riconoscimento di oggetti in un'immagine, con tempi di allenamento e dataset ridotti. Nonostante il contensto di sviluppo originale, la UNet è stata utilizzata e studiata in una moltitudine di campi diversi, spaziando dal settore agricolo [9] al telerilevamento [10]. La rete è completamente convoluzionale (fully convolutional neural network), ovvero non include utilizzo di layer densi.

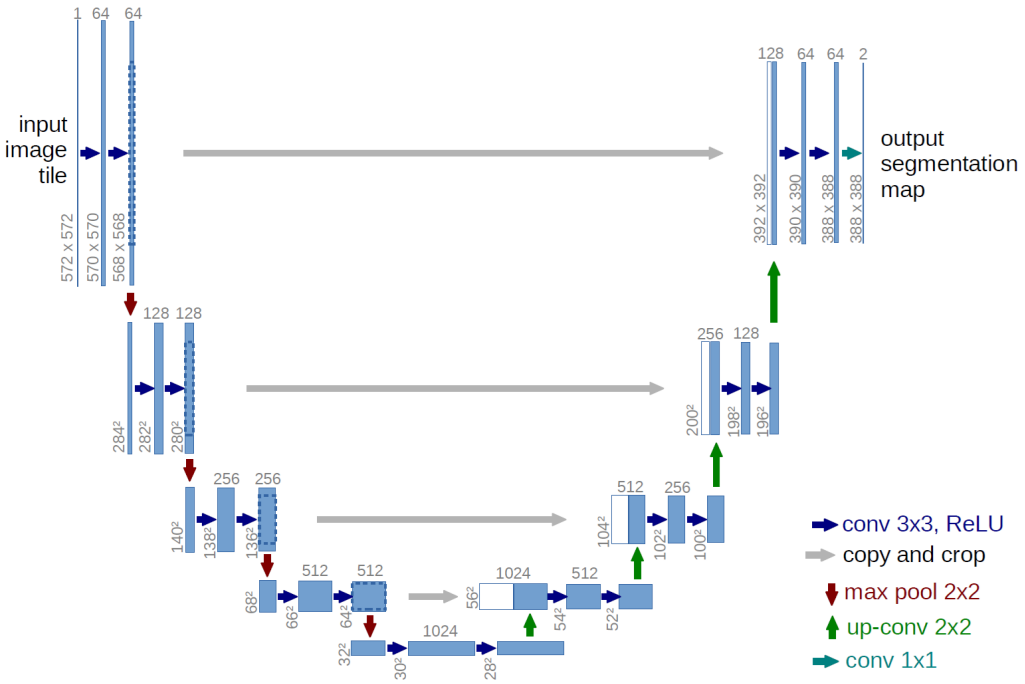


Figura 1.5: Architettura UNet originale

L'architettura originale della UNet (visibile nella Figura 1.5) è caratterizzata da una forma simmetrica, composta da 2 sezioni principali: la contrazione e l'espansione. La sezione di contrazione ha il compito di estrarre le caratteristiche più rilevanti dell'immagine, riducendo la sua dimensione spa-

ziale tramite layer di MaxPooling. Contrariamente la sezione di espansione si occupa di concatenare le informazioni estratte dai livelli inferiori, con le feature map provenienti dalla corrispettiva parte di encoding. Per questo scopo, vengono utilizzati layer di upsampling, che aumentano la dimensione spaziale dell'immagine, riducendo il numero di feature map.

1.3 Interpolazione delle proiezioni per la tomografia a viste sparse

Ribadendo lo scopo di questa tesi, l'obiettivo è stato quello di aumentare il numero di proiezioni di uno stack CBCT sparse-view, al fine di migliorare la qualità delle ricostruzioni 3D tramite il metodo FDK. Per maggiore chiarezza, si denoterà con n il numero delle proiezioni sparse in input, mentre con N si indicherà il numero di proiezioni in output dalle tecniche di image processing. Considereremo sempre N come multiplo di n corrispondente a una potenza di 2, in modo tale da preservare le proiezioni originali nello stack di proiezioni aumentate. Per farlo si sono impiegate tecniche di interpolazione di immagini, ovvero tecniche per stimare il valore di alcuni punti sconosciuti a partire da dati noti. Nello specifico il task perseguito è stato quello di utilizzare due proiezioni, intese come immagini 2D da due angolazioni prossime fra loro, per predire la proiezione mancante fra queste due.

1.3.1 Metodi naive per interpolazione

In letteratura sono conosciuti molteplici metodi naive per l'interpolazione e l'inpainting su immagini. Per utilizzare questi approcci in questo specifico task è stato necessario cambiare visione del problema, infatti canonicamente questi approcci ricostruiscono porzioni di un'immagine 2D, senza basarsi su immagini multiple. Per farlo si è modificato lo stack di proiezioni originali aggiungendo una proiezione vuota fra ogni coppia di immagini presenti. Successivamente, analizzando lo stack come pila verticale di immagini, si sono

divise le proiezioni per colonna, ottenendo in output immagini composte da righe originali e vuote alternate. Questo ha permesso di applicare gli algoritmi in quanto l'input risultava essere una comune immagine 2D. Purtroppo l'esito di queste predizioni risultava essere molto approssimativo e impreciso, richiedendo l'utilizzo di tecniche più efficaci.

1.3.2 Metodi di Deep Learning per interpolazione

L'utilizzo di reti neurali per task di interpolazione può migliorare di molto i risultati, anche per via dell'allenamento specifico sul problema stesso. Infatti la possibilità di ridurre il problema a certe tipologie di immagini può aiutare il modello a riconoscere pattern comuni, e quindi correlare porzioni delle due immagini di input in maniera più "consapevole". Questo comportamento spesso migliora significativamente i dettagli delle immagini, perchè la rete sarà maggiormente incline alla generazione di forme più adatte al contesto e quindi più naturali.

In letteratura si sono già esplorate simili applicazioni per tecnologie fan-beam [11], dimostrando l'efficacia di questi modelli per il miglioramento di ricostruzioni sparse-view.

1.4 Contributo della tesi

In questa tesi si approfondisce l'utilizzo di reti neurali interpolative a supporto delle tecniche di sparse-view CBCT. Lo scopo primario è migliorare la qualità delle ricostruzioni tomografiche, aumentando la quantità di proiezioni disponibili per l'algoritmo FDK. Si vogliono inoltre mantenere inalterate le proiezioni originali a disposizione, per una maggiore precisione nei dettagli durante la ricostruzione.

Capitolo 2

Metodo proposto

In questo capitolo viene presentato l'approccio proposto e illustrati i metodi utilizzati per la sua implementazione.

2.1 Rete neurale interpolativa

Dati i chiari limiti delle metodologie naive e i potenziali vantaggi dei metodi di deep learning analizzati in precedenza, si è impiegato l'utilizzo di una rete neurale a scopo di interpolazione.

2.1.1 Approccio al problema

Ribadendo l'obiettivo da perseguire, si desidera realizzare un metodo per aumentare la quantità di proiezioni da n a N, dove N è un multiplo di n corrispondente a una potenza di 2. Al fine di realizzare una rete neurale più versatile, leggera e precisa nei dettagli, si è deciso di ridurre il problema ad un problema più contenuto. Infatti si è scelto di realizzare un modello per la generazione di una singola proiezione mancante per volta, sulla base di alcune proiezioni vicine. La proiezione da generare è stata fissata essere sempre quella esattamente centrale fra le proiezioni fornite in input. Numericamente, se le proiezioni in input dovessero essere collocate a 10 gradi di distanza angolare fra loro, la rete ricostruirebbe una proiezione posizionata

al loro centro, ovvero a 5 gradi da entrambe. In questo modo risulta possibile applicare n volte la rete neurale allo stack in input per ottenere uno stack con $2n$ proiezioni, come mostrato in Figura 2.1. È successivamente possibile riapplicare $2n$ volte la rete allo stack ottenuto per ottenere $4n$ proiezioni e così via.

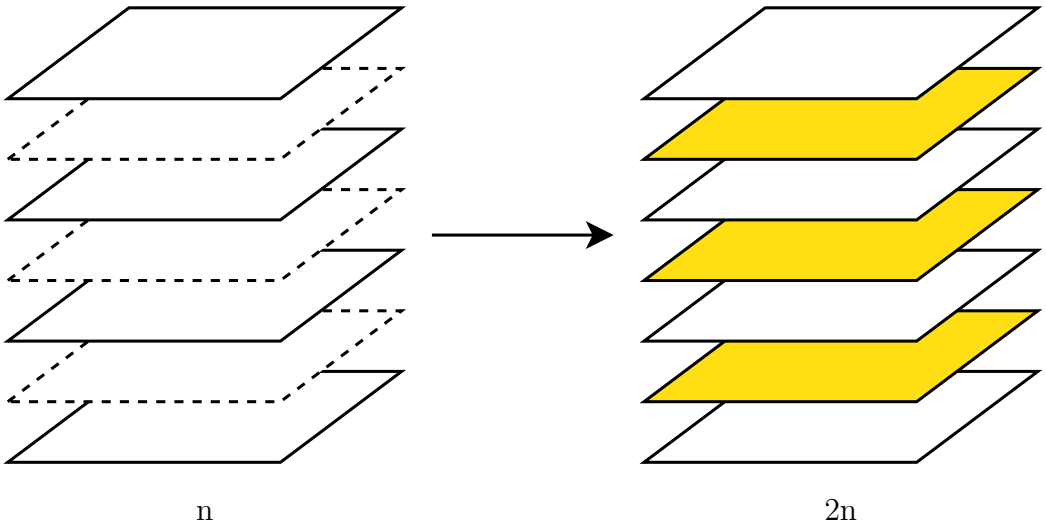


Figura 2.1: Incremento delle proiezioni di uno stack

Per questo motivo, per maggiore semplicità, si considererà solamente il caso $N = 2n$, ricordando che essendo il processo ripetibile, è possibile raggiungere una qualsiasi potenza di 2. Infine tramite questo approccio, è possibile garantire che, dopo ogni applicazione della rete, le proiezioni precedenti rimangano tutte presenti e invariate.

2.1.2 Architettura della rete

Come architettura del modello si è realizzata una variante della precedentemente analizzata UNet, dove i principali blocchi convolutivi sono stati rimpiazzati da blocchi convolutivi residuali, realizzando una UNet residuale. Infatti analizzando lo scopo principale del progetto, si può notare come la maggior parte delle immagini prodotte siano estremamente simili alle origi-

nali, con variazioni derivate dal cambio di angolazione. Per questo motivo, un approccio residuale risulta maggiormente indicato, considerato che, tramite esso, la rete inizierà ad apprendere il residuo rispetto all'immagine originale, piuttosto che tentare di ricostruire grandi porzioni di essa. Con residuo si intende la variazione da applicare all'immagine originale per l'ottenimento di quella target. Questa modifica punta quindi ad ottimizzare la rete per lo specifico task prefissato, oltre a ridurre i tempi di training. Un'altra modifica applicata è stata l'introduzione di un bottleneck dilatato, ovvero una variante del blocco convolutivo standard in cui il campo visivo (receptive field) è aumentato "saltando" dei pixel dell'immagine analizzata. Questa modifica è stata effettuata per identificare pattern avanzati che riguardassero porzioni più grandi dell'immagine, rispetto alla visione normale del kernel scelto. Inoltre questo approccio ha permesso di mantenere un numero di parametri estremamente più contenuto. Un'altra differenza introdotta è stata la riduzione del numero di feature map. Inizialmente infatti si erano mantenute 64 feature map, al fine di estrarre quante più informazioni possibili dalle immagini, ma questo risultava un bilanciamento non ottimale fra qualità delle feature e batches in memoria. Infatti la precedente configurazione richiedeva un oneroso quantitativo di memoria per l'allenamento, seppur l'utilizzo della componente di calcolo risultasse poco utilizzata. Aumentando però la quantità di immagini processate contemporaneamente, e riducendo le feature map a 48, è stato possibile aumentare anche il carico di lavoro sulla GPU, sfruttando al meglio tutte le risorse hardware a disposizione. Sono stati infine sostituiti i layer di MaxPooling con layer di AvgPooling.

Layer di normalizzazione

Al fine di migliorare la stabilità di allenamento e le prestazioni del modello, sono stati inseriti alcuni layer di normalizzazione. Inizialmente è stato introdotto l'utilizzo di alcuni layer di BatchNorm, che permettono di ridurre la variabilità statistica delle attivazioni all'interno di ciascun batch, favorendo una convergenza più rapida e riducendo il rischio di problemi legati al

vanishing o all'exploding gradient. Un problema intrinseco legato all'utilizzo di BatchNorm risiede proprio nel metodo di calcolo utilizzato per la normalizzazione, infatti basandosi direttamente sugli elementi del batch risulta particolarmente soggetto ad oscillazioni in caso di batch più piccoli. Per questo motivo sono stati sostituiti i layer di BatchNorm con layer di GroupNorm indipendenti dalla dimensione del batch. Il numero di gruppi utilizzati per ogni layer di normalizzazione è stato selezionato come il massimo valore fra $(32, 16, 8, 4, 2)$, che dividesse perfettamente il numero di canali in input e che mantenesse almeno 4 canali per ogni gruppo. Il limite posto ad almeno 4 canali per ogni gruppo è finalizzato al mantenimento di una media e varianza più stabili.

2.1.3 Loss function

Come funzione di comparazione fra target atteso e immagine predetta, si è utilizzata una combinazione fra due funzioni note in letteratura, quali *Charbonnier* e l'indicatore *MS-SSIM*. La funzione di perdita di Charbonnier rappresenta una variante robusta della tradizionale *L2 loss*, ed è particolarmente adatta a ridurre l'effetto di outlier durante l'addestramento [12, 13]. Essa è definita come:

$$\mathcal{L}_{Charbonnier}(x, y) = \sqrt{(x - y)^2 + \epsilon^2} \tag{2.1}$$

dove x rappresenta il valore del pixel predetto, y il valore del pixel target e ϵ è un piccolo termine costante che evita la singolarità della radice quadrata, garantendo stabilità numerica. Tale funzione penalizza in maniera morbida le discrepanze tra predizione e target, risultando più robusta rispetto alla semplice differenza quadratica.

L'*MS-SSIM* (*Multi-Scale Structural Similarity Index* [14]) è un indicatore volto a catturare la similarità strutturale tra due immagini su più scale di risoluzione, riflettendo meglio la percezione visiva rispetto alle normali metriche di errore pixel-wise. La definizione di MS-SSIM è data da:

$$\text{MS-SSIM}(x,y) = [l_M(x,y)]^{\alpha_M} \cdot \prod_{j=1}^{M} [c_j(x,y)]^{\beta_j} [s_j(x,y)]^{\gamma_j}, \tag{2.2}$$

dove l_M è il termine di luminanza, calcolato all'ultima scala M, c_j e s_j rappresentano rispettivamente i termini di contrasto e struttura alla scala j, mentre α_M, β_j e γ_j sono pesi che bilanciano l'influenza di ciascun termine. L'indice MS-SSIM assume valori compresi tra 0 e 1, dove valori più prossimi a 1 indicano una maggiore similarità strutturale.

Volendo ottenere una funzione di perdita da minimizzare durante l'addestramento, il valore dell'MS-SSIM è stato invertito in questo modo:

$$\mathcal{L}_{MS\text{-}SSIM}(x,y) = 1 - \text{MS-SSIM}(x,y) \tag{2.3}$$

in modo che valori più elevati della perdita corrispondano a una minore similarità tra immagine predetta e target.

Infine, entrambe le componenti descritte sono state pesate differentemente e unite in un'unica funzione di loss:

$$\mathcal{L}(x,y) = \mathcal{L}_{Charbonnier}(x,y) - 0.4 \cdot \mathcal{L}_{MS\text{-}SSIM}(x,y) \tag{2.4}$$

2.2 Varianti del modello

Avendo a disposizione come input uno stack da n proiezioni, si sono realizzate alcune varianti del modello che basassero le loro previsioni su un diverso numero di proiezioni stesse. Si farà riferimento a questa differenza fra i modelli, utilizzando l'espressione "c angoli di visione". È importante notare come questa nomenclatura indichi comunque una porzione contigua di proiezioni provenienti dallo stack in input. Tramite l'aumento degli angoli di visione si mira a fornire un maggiore contesto spaziale alla rete, in modo tale che le proiezioni in output vengano generate sulla base di maggiori dettagli spaziali. Per questo motivo, in questa analisi, ci si è soffermati su 2 varianti principali, ovvero i modelli con 2 e 4 angoli di visione, i cui risultati sono presentati e confrontati nel Capitolo 4.

Per via di queste varianti, il dataset è stato diviso in unità dette campioni (sample), di quantità e dimensione variabile in base alla quantità di angoli di visione. All'interno di ogni campione sono contenute tutte le proiezioni da fornire in input, per ottenere come risultato l'immagine predetta. Generalmente, gli input completi da fornire alle reti seguono questa forma:

$$(batches, channels, height, width) \tag{2.5}$$

Analizzando queste componenti troviamo:

- **batches:** indica la quantità di campioni passati alla rete. Questi verranno elaborati parallelamente e forniranno in output una quantità **batches** di risultati.
- **channels:** indica il numero di proiezioni per ogni campione. Numericamente possono essere 2 o 4 in base al modello preso in esame.
- **height:** altezza delle proiezioni (400 pixel).
- **width:** larghezza delle proiezioni (400 pixel).

2.2.1 Indicizzazione delle proiezioni

Come descritto precedentemente, la costruzione del problema ci garantisce che qualsiasi proiezione proveniente dallo stack in input da n proiezioni sarà preservata nello stack di output da N proiezioni. Grazie a questa proprietà, e per fini di leggibilità, si utilizzerà unicamente lo stack da N proiezioni per riferirsi sia alle proiezioni in input, che alle proiezioni in output. Nello specifico si utilizzerà l'indice i, nell'insieme $\{0, 1, 2, \ldots, N-1\}$ per riferirsi alla proiezione da generare.

2.2.2 Modello a 2 angoli di visione

Il modello considerato standard è stato fissato avere 2 angoli di visione, ovvero utilizzare una proiezione a sinistra e una a destra rispetto a quella da

predire (esempio in Figura 2.2). Numericamente come input sono fornite le proiezioni:

$$\{i-1, i+1\} \tag{2.6}$$

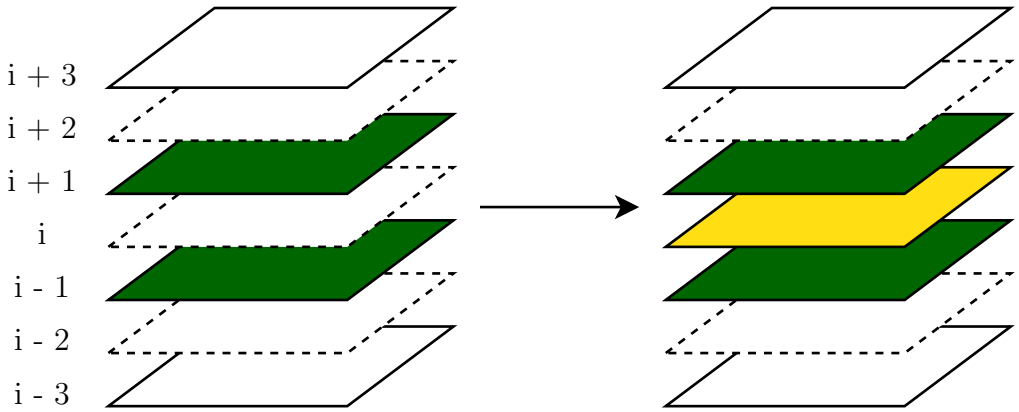


Figura 2.2: Generazione di una proiezione mancante, sulla base di 2 angoli di visione

2.2.3 Modello a 4 angoli di visione

Differentemente utilizzando il modello con 4 angoli di visione, sarà necessario escludere dall'input tutte le proiezioni non coerenti al problema da risolvere (esempio in Figura 2.3). Numericamente come input sono fornite le proiezioni:

$$\{i-3, i-1, i+1, i+3\} \tag{2.7}$$

Il motivo dell'assenza delle proiezioni $i-2$ e $i+2$ è relativa al fatto che queste proiezioni non provengono dallo stack di input. Infatti, utilizzando lo stack di output (da N proiezioni) come riferimento, esse saranno proiezioni target, quindi non presenti nello stack originale. Se non si escludessero le proiezioni indicate, la rete proseguirebbe l'allenamento su proiezioni non coerenti rispetto all'input utilizzato successivamente in inferenza.

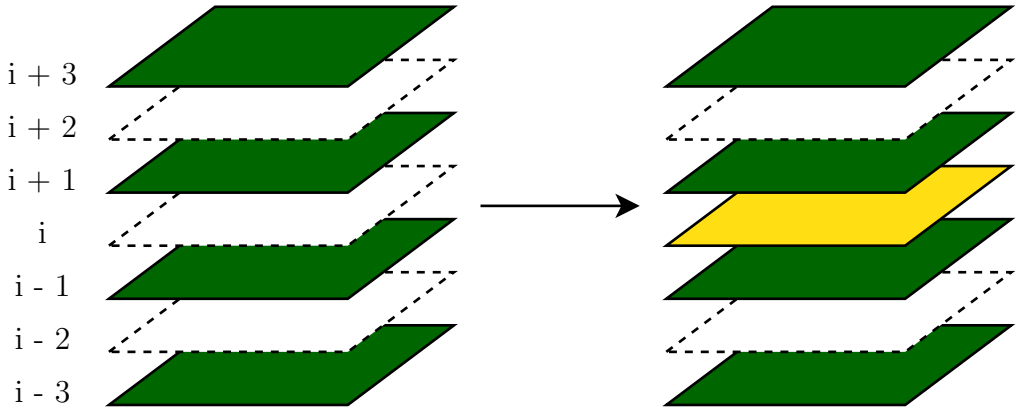


Figura 2.3: Generazione di una proiezione mancante, sulla base di 4 angoli di visione

Infatti, la distanza angolare fra $i-3$ e $i-1$ risulta essere la stessa distanza presente nello stack con n proiezioni, mentre la distanza fra $i-2$ e $i-1$ risulta dimezzata.

Capitolo 3

Descrizione degli esperimenti

In questa sezione si presentano gli esperimenti condotti, i cui risultati saranno presentati nel Capitolo 4.

3.1 Dataset tomografici

L'insieme dei dati di partenza è composto da 100 volumi 3D, uno per ogni file, di dimensione $(256, 256, 256)$. Ogni volume dell'insieme è generato in modo sintetico tramite un algoritmo pseudo-casuale. Partendo da un volume completamente vuoto, esso inserisce in varie parti del volume degli elementi di ostacolo ai raggi, simulando la presenza di strutture ossee frammentate in varie parti. Le principali strutture inserite sono ellissoidi, sfere e cilindri sottili. Alcuni esempi di volumi generati sono mostrati nella Figura 3.1.

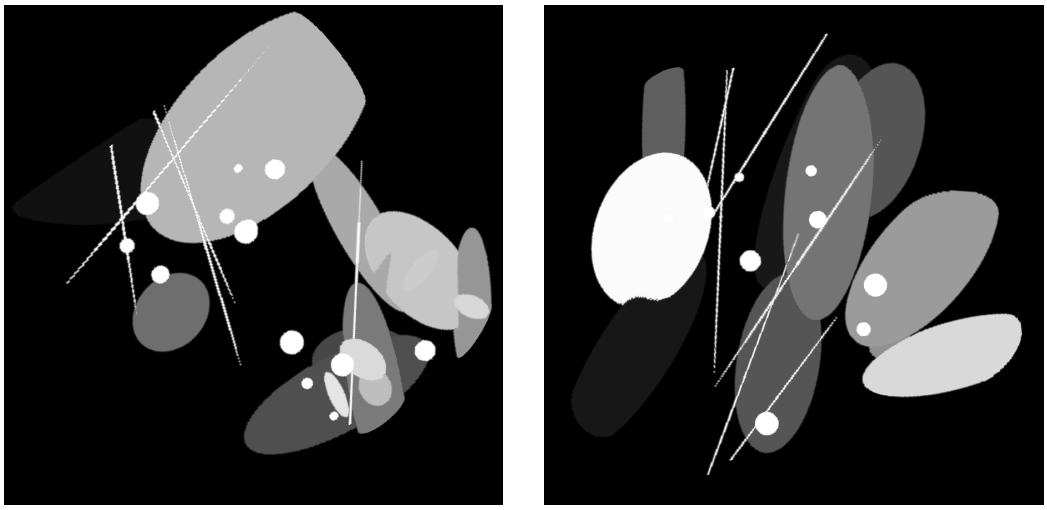
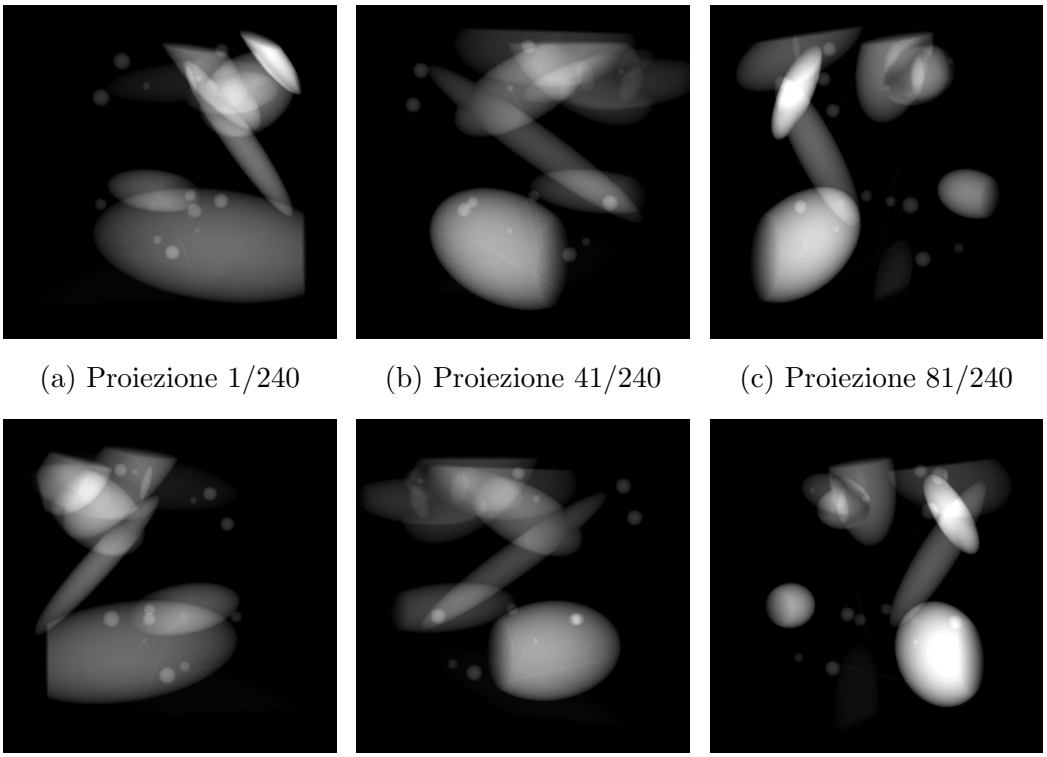


Figura 3.1: Esempi di volumi originali, con vari elementi di ostacolo ai raggi

Una volta ottenuti i volumi è stato possibile generare il dataset, realizzando delle proiezioni simulate sui volumi stessi. Il dataset delle proiezioni è stato generato fissando la distanza angolare a 1.5 gradi, l'intervallo angolare a $[0, 360]$ gradi e il numero di proiezioni a $N = 240$ per ogni volume. Per la realizzazione delle proiezioni simulate sui volumi si è impiegato l'utilizzo della libreria ASTRA Toolbox [15], una libreria Python specializzata in operazioni di tomografia, con supporto all'accelerazione hardware su GPU ad alte performance.

La libreria è stata impiegata per la simulazione di un fascio conico CBCT con queste impostazioni:

- **Distanza della sorgente dal volume:** 1000.0 mm
- **Distanza del detector del volume:** 200.0 mm
- **Dimensione del pixel del rilevatore X:** 1.0 mm
- **Dimensione del pixel del rilevatore Y:** 1.0 mm
- **Numero di pixel del rilevatore X:** 400
- **Numero di pixel del rilevatore Y:** 400



(a) Proiezione 1/240 (b) Proiezione 41/240 (c) Proiezione 81/240

(d) Proiezione 121/240 (e) Proiezione 161/240 (f) Proiezione 201/240

Figura 3.2: Visualizzazione di 6 proiezioni originali di un volume

La Figura 3.2 mostra alcune proiezioni dello stesso volume da diversi angoli di scansione.

Per praticità e organizzazione le proiezioni sono state così divise: ogni volume originale veniva elaborato e ne venivano costruite $N = 240$ proiezioni, queste ultime sono poi state impilate in forma matriciale sull'asse Z. In questo modo è stato possibile il salvataggio in un unico file di tutte le proiezioni del singolo volume. Per questo motivo la dimensione del dataset è rimasta di 100 file.

L'obiettivo principale considerato in questa tesi è di aumentare le proiezioni da $n = 120$ a $N = 240$, quindi raddoppiare i dati tomografici. Tuttavia è stato anche considerato il caso con $n = 120$ e $N = 480$, nel quale si sono mantenuti invariati l'intervallo angolare di $[0, 360]$ gradi e la modalità di si-

mulazione con ASTRA. I corrispondenti risultati sono mostrati nella Sezione 4.2.

3.2 Allenamento delle reti

L'allenamento delle reti è stato interamente effettuato su schede video dedicate *RTX 2080 Ti* con 11 GB di memoria VRAM. Per mantenere il quantitativo di memoria utilizzata all'interno dei limiti, e per massimizzare l'utilizzo delle risorse hardware, è stato necessario implementare alcune accortezze nella componente di allenamento.

Caricamento dei campioni

Il metodo di caricamento dei campioni in memoria è stato migliorato al fine di ottenere le massime performance dall'hardware disponibile.

Inizialmente il caricamento avveniva attraverso l'implementazione di una finestra scorrevole, che selezionava un piccolo numero di file di proiezioni per volta e li divideva in campioni. Quando tutti i campioni di un file venivano processati, esso veniva deallocato e sostituito con un nuovo file. Questo permetteva di caricare in differita i file del dataset da disco, senza interrompere l'allenamento.

Purtroppo questo approccio introduceva un bias durante l'apprendimento, poichè le proiezioni fornite provenivano tutte dallo stesso file e da angolazioni molto simili fra di loro. Per questo motivo è stato necessario introdurre un approccio che permettesse lo shuffle dei campioni, in modo che provenissero da un punto casuale del dataset.

Per farlo, si è utilizzato il caricamento selettivo di una proiezione a partire da un file, leggendo solo la porzione di esso che conteneva la proiezione desiderata. Tramite quindi un sistema di indici che identificasse univocamente una singola proiezione in tutto il dataset, è stato possibile implementare una scelta casuale della proiezione durante il training.

Nonostante la migliorata velocità di caricamento dei dati, questa operazione avveniva in modo sequenziale durante l'allenamento, introducendo tempi di attesa. Per questo motivo infine, si è scelto di mantenere un buffer di 10 campioni sempre disponibili, caricandoli in differita durante il training quando ne venivano utilizzati.

Ottimizzatore

Come ottimizzatore, utilizzato per minimizzare la loss (2.4), si è utilizzato il consolidato *AdamW*, una variante di *Adam* maggiormente stabile e con una regolarizzazione più efficace. I vantaggi di questi ottimizzatori risiedono principalmente nella capacità di convergenza rapida e stabile, nella buona resistenza al rumore e nel calcolo adattivo del learning rate su ciascun parametro.

La differenza fra i 2 risiede nell'utilizzo del weight decay, una tecnica di regolarizzazione che mira a limitare la crescita incondizionata dei pesi della rete. Tramite di esso è possibile mitigare parzialmente l'overfitting sui dati e migliorare la generalizzazione.

Specificatamente i due ottimizzatori differiscono per la modalità di applicazione di questa regolarizzazione. In *Adam*, il weight decay viene applicato direttamente al gradiente prima dell'aggiornamento dei pesi, comportando il fatto che la penalizzazione possa venir influenzata dal meccanismo adattivo dell'ottimizzatore, il quale normalizza e scala i gradienti in base ai momenti dei parametri.

Questo comporta una differenza sull'effetto della regolarizzazione rispetto al weight decay teorico, risultando meno uniforme e prevedibile.

AdamW corregge questa limitazione separando i due processi. Primariamente eseguendo l'aggiornamento dei pesi tramite *Adam* e successivamente applicando il decadimento direttamente ai parametri, preservando l'effetto originale della regolarizzazione e garantendo maggiore stabilità e capacità di generalizzazione [16].

Learning rate

Per adattare dinamicamente il learning rate, quando necessario, durante l'allenamento, è stata introdotta una componenente *Reduce on Plateau*, partendo da un valore base di 10^{-3}. Le condizioni per la riduzione del valore sono state impostate ad un decremento del 50% del learning rate dopo 3 epoche senza miglioramenti. Per evitare una riduzione troppo eccessiva del learning rate, si è poi impostato il valore minimo possibile a 10^{-6}.

Parametri di allenamento

Per l'allenamento delle reti, il dataset è stato diviso in 2 sotto-dataset più ridotti. Il primo è stato impiegato nell'allenamento della rete, includendo il 90% dei file, mentre il secondo solo per la validazione, implementata per prevenire overfitting. La dimensione del batch è stata fissata a 8 campioni in contemporanea e il numero di epoche è stato fissato ad un massimo di 80. Infine sono stati generati 240 campioni per ogni file, per un totale di 24.000 campioni totali, di cui 21.600 di allenamento.

Capitolo 4

Risultati

In questo capitolo si riportano ed analizzano i risultati ottenuti.

4.1 Esperimenti da 120 a 240 viste

In questa sezione si presentano gli esperimenti effettuati per l'aumento delle proiezioni da 120 a 240 viste, comparando i risultati con l'interpolazione lineare.

4.1.1 Fase di addestramento

L'addestramento è stato effettuato mantenendo i parametri di allenamento equivalenti fra i due modelli. Tuttavia, è possibile identificare una importante differenza fra i due training, relativa al numero di epoche raggiunte. Il principale motivo di questa differenza è derivato da un maggior impegno computazionale investito per il modello con 4 angoli di visione. Esso infatti, processando più valori di input, richiede più potenza di calcolo e quindi più tempo.

Il numero di epoche raggiunte risulta quindi minore a causa di un timeout pre-impostato nel job di allenamento, fissato a 24 ore per entrambi i modelli. Tuttavia questa differenza, essendo localizzata in sezioni finali del training, presenta differenze percentuali non particolarmente significative.

Sono infine riportati in Figura 4.1 e in Figura 4.2 i grafici di allenamento dei due modelli, i quali presentano un confronto fra i valori di loss sul set di allenamento e sul set di validazione.

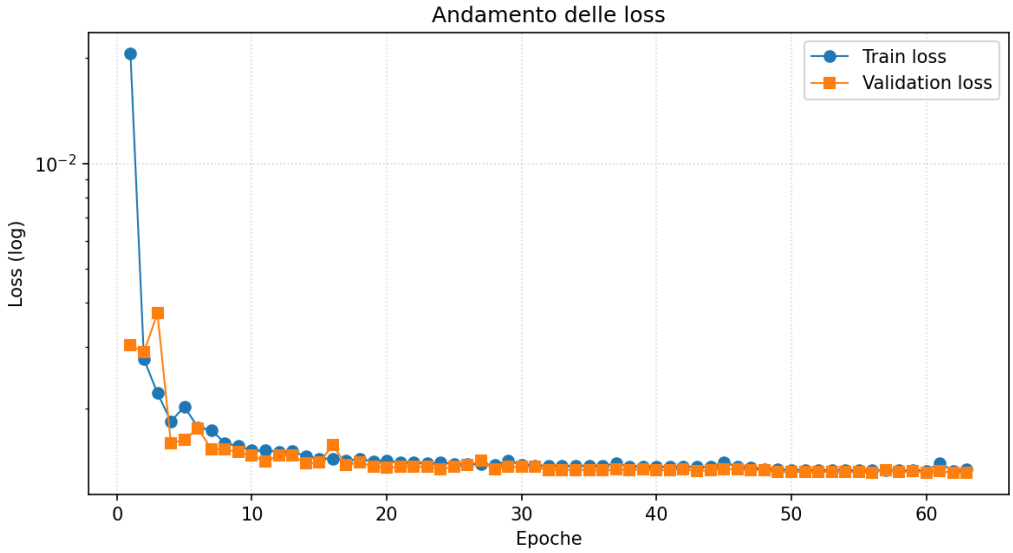


Figura 4.1: Andamento delle loss del modello con 2 angoli

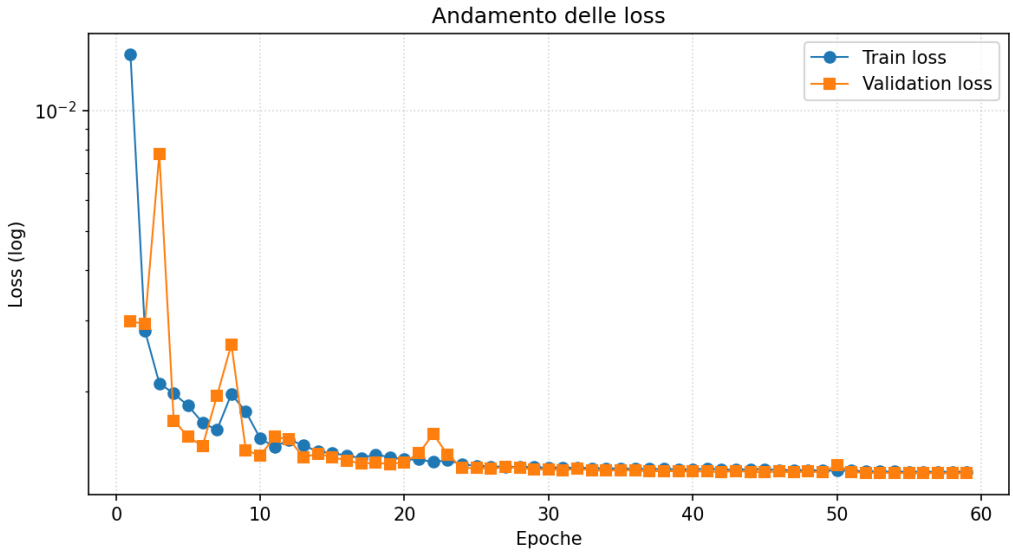
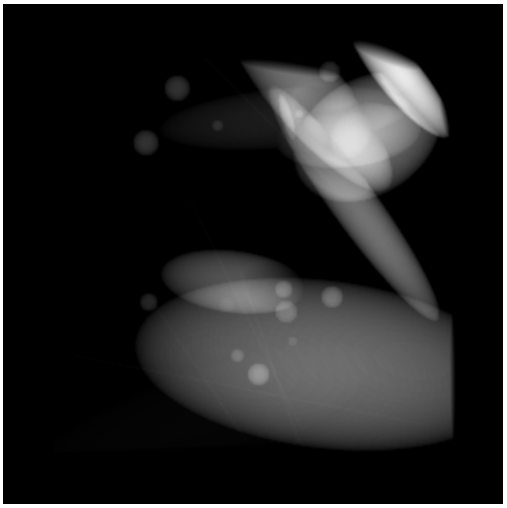


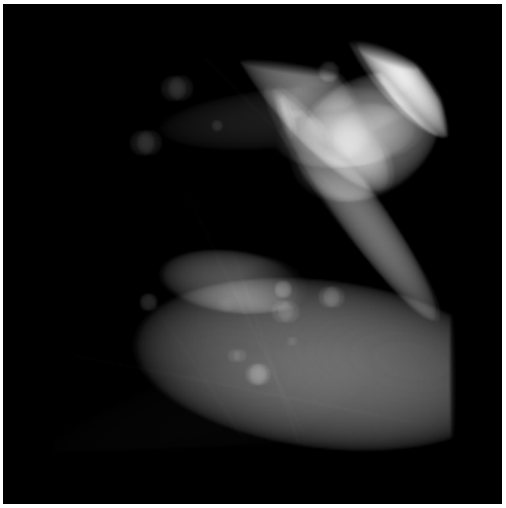
Figura 4.2: Andamento delle loss del modello con 4 angoli

4.1.2 Inferenza sulle proiezioni

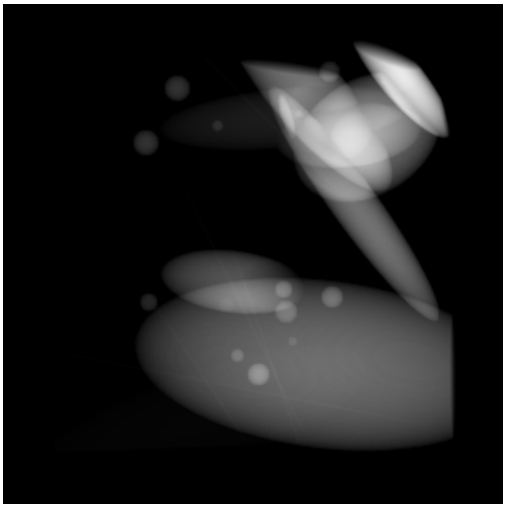
Per verificare le prestazioni dei modelli è stata eseguita l'inferenza su 10 file, non facenti parte del dataset di allenamento, e ne sono stati stilati i risultati in comparazione con le proiezioni originali.



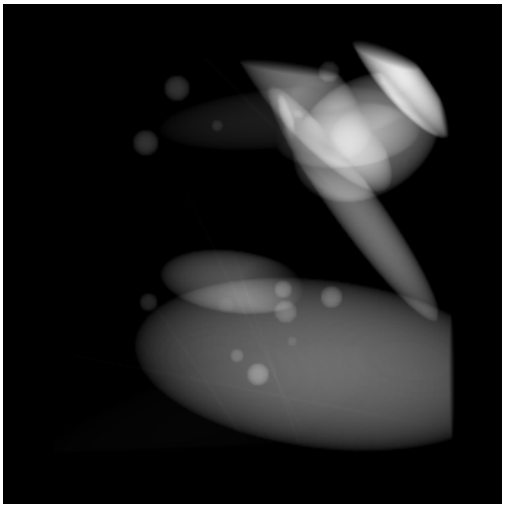
(a) Proiezione originale



(b) Proiezione interp. lineare



(c) Proiezione modello 2 angoli



(d) Proiezione modello 4 angoli

Figura 4.3: Confronto fra 4 proiezioni realizzate con diverse tecniche

Un esempio di confronto fra proiezioni è presente nella Figura 4.3, nella

quale è possibile confrontare visivamente i risultati delle varie tecniche prese in esame. Tramite essa è possibile notare come la tecnica naive introduca una significativa perdita di dettaglio e di precisione nella ricostruzione dell'immagine. Contrariamente è possibile notare come entrambi i modelli producano proiezioni estremamente simili all'originale, rappresentando un risultato più definito, come supportato dai dati sulle metriche riportate nelle Tabelle 4.1a, 4.1b, 4.1c.

Metrica	Mean	Std	Min	Max	Median
MSE	0.2132	0.0743	0.1264	0.3657	0.2031
MAE	0.0854	0.0167	0.0605	0.1199	0.0828
RMSE	0.4553	0.0769	0.3555	0.6048	0.4504
PSNR	55.0838	1.4193	52.4992	57.1149	55.0628
SSIM	0.9987	0.0003	0.9980	0.9992	0.9988
NCC	0.9996	0.0002	0.9991	0.9998	0.9997
SNR	32.5629	1.8067	28.8035	35.0406	33.1759

(a) Risultati delle metriche con interpolazione lineare

Metrica	Mean	Std	Min	Max	Median
MSE	0.0092	0.0028	0.0055	0.0155	0.0088
MAE	0.0261	0.0048	0.0200	0.0364	0.0250
RMSE	0.0947	0.0139	0.0741	0.1245	0.0941
PSNR	68.6950	1.2520	66.2276	70.7368	68.6636
SSIM	0.9999	0.0000	0.9999	0.9999	0.9999
NCC	1.0000	0.0000	1.0000	1.0000	1.0000
SNR	46.1741	1.3872	43.7196	48.2169	46.4506

(b) Risultati delle metriche con il modello a 2 angoli

Metrica	Mean	Std	Min	Max	Median
MSE	0.0093	0.0031	0.0055	0.0168	0.0088
MAE	0.0261	0.0049	0.0198	0.0369	0.0249
RMSE	0.0954	0.0151	0.0740	0.1298	0.0937
PSNR	68.6445	1.3225	65.8655	70.7470	68.7033
SSIM	0.9999	0.0000	0.9999	0.9999	0.9999
NCC	1.0000	0.0000	1.0000	1.0000	1.0000
SNR	46.1236	1.4620	43.5022	48.2271	46.4433

(c) Risultati delle metriche con il modello a 4 angoli

Tabella 4.1: Risultati delle proiezioni generate, rispetto alle originali

Dai risultati ottenuti è possibile notare un notevole miglioramento fra i risultati naive e i risultati dei modelli. Questo è principalmente dovuto alla natura dell'interpolazione lineare. Essa infatti, ricercando una proiezione posizionata esattamente al centro fra le 2, realizza di fatto una media, non tenendo in considerazione la struttura degli oggetti presenti nel volume.

Metrica	**Interp. Lineare**	**2 angoli**	**Miglioramento (%)**
MSE	0.2132	0.0092	+95.7%
MAE	0.0854	0.0261	+69.4%
RMSE	0.4553	0.0947	+79.2%
PSNR	55.0838	68.6950	+24.7%
SSIM	0.9987	0.9999	+0.1%
NCC	0.9996	1.0000	+0.0%
SNR	32.5629	46.1741	+41.8%

(a) Confronto metriche medie fra interpolazione naive e modello a 2 angoli

Metrica	**Interp. Lineare**	**4 angoli**	**Miglioramento (%)**
MSE	0.2132	0.0093	+95.6%
MAE	0.0854	0.0261	+69.5%
RMSE	0.4553	0.0954	+79.0%
PSNR	55.0838	68.6445	+24.6%
SSIM	0.9987	0.9999	+0.1%
NCC	0.9996	1.0000	+0.0%
SNR	32.5629	46.1236	+41.6%

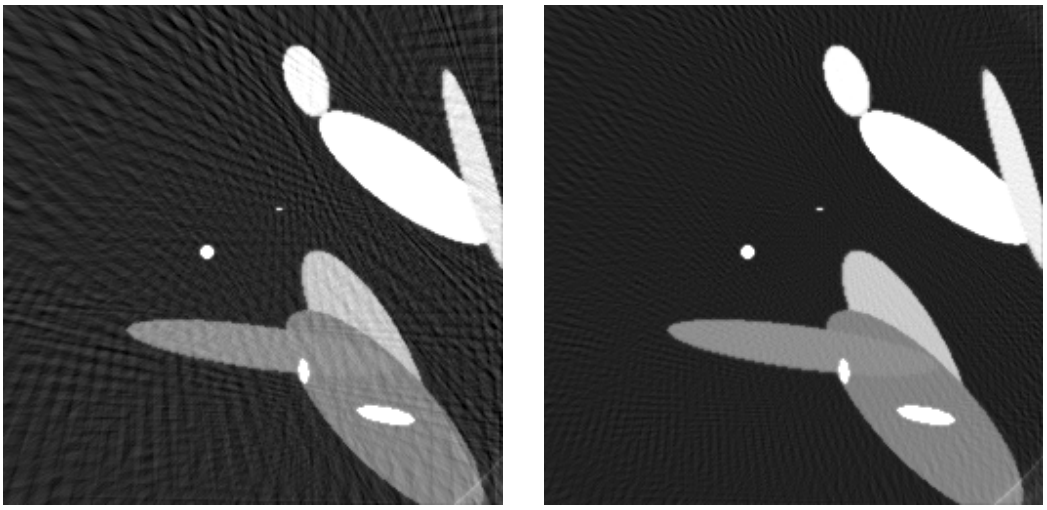
(b) Confronto metriche medie fra interpolazione naive e modello a 4 angoli

Tabella 4.2: Confronti sulle metriche medie delle proiezioni

Tramite le Tabelle 4.2a e 4.2b si confrontano i risultati medi per ogni metrica, ed è possibile notare come il maggior numero di angoli di visione non corrisponda strettamente ad un maggiore incremento prestazionale. Infatti il modello con 4 angoli di visione non mostra differenze significative nelle statistiche considerate.

4.1.3 Ricostruzione dei volumi

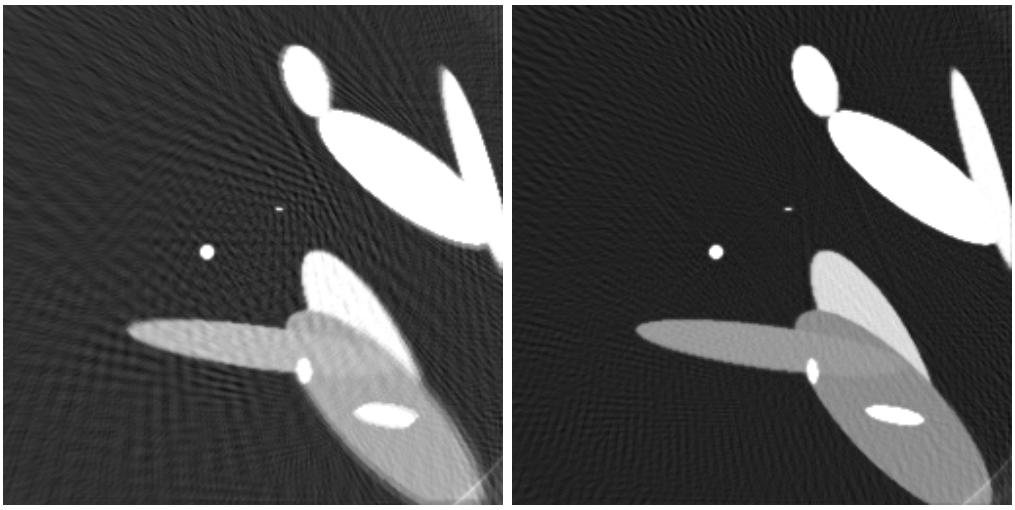
A fini comparativi si sono riportate due slice (Figura 4.4), provenienti rispettivamente dai volumi ricostruiti con 120 e 240 proiezioni originali. Questi volumi sono stati ricostruiti senza l'intervento di alcuna tecnica di aumento delle proiezioni. Tramite di esse possiamo notare come il maggiore numero di proiezioni riduca significativamente gli artefatti grafici. Ciononostante, è comunque possibile notare alcuni artefatti nel volume ricostruito con 240 proiezioni, seppur estremamente ridotti.



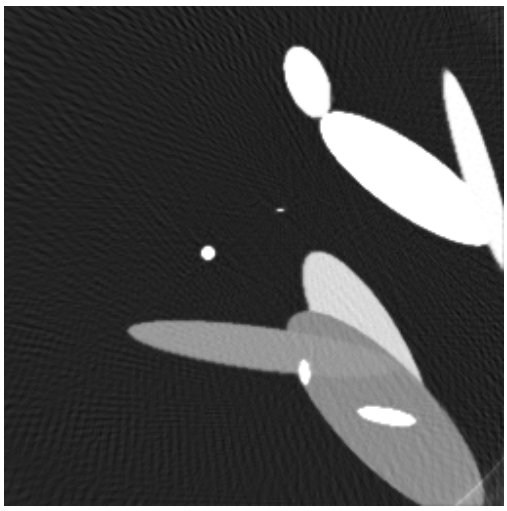
(a) 120 originali (b) 240 originali

Figura 4.4: Confronto fra volumi ricostruiti tramite sole proiezioni originali

Attraverso la Figura 4.5 si riportano alcuni confronti fra slice, provenienti da volumi ricostruiti tramite stack di proiezioni interpolate. Tramite essi è possibile notare alcune somiglianze rispetto a quanto già visto nel confronto fra proiezioni. Il modello con 4 angoli di visione infatti produce una ricostruzione drasticamente migliore dell'approccio naive, senza evidenti differenze rispetto al modello a 2 angoli di visione.



(a) Interpolazione lineare (b) Modello 2 angoli



(c) Modello 4 angoli

Figura 4.5: Confronto fra volumi ricostruiti con proiezioni interpolate, realizzate con diverse tecniche

Anche in questo caso si presentano i dati aggregati di tutte le ricostruzioni nelle Tabelle 4.3a, 4.3b, 4.3c, prendendo come base del confronto i volumi ricostruiti con 240 proiezioni completamente originali.

Metrica	Mean	Std	Min	Max	Median
MSE	$4.18 \cdot 10^{-4}$	$1.16 \cdot 10^{-4}$	$2.78 \cdot 10^{-4}$	$6.63 \cdot 10^{-4}$	$4.13 \cdot 10^{-4}$
MAE	0.0118	0.0018	0.0092	0.0156	0.0117
RMSE	0.0203	0.0028	0.0167	0.0258	0.0203
PSNR	33.9417	1.1612	31.7828	35.5587	33.8450
SSIM	0.8578	0.0274	0.8025	0.8932	0.8631
NCC	0.9928	0.0019	0.9885	0.9953	0.9931
SNR	19.0432	1.1594	16.7560	20.6579	19.1078

(a) Risultati delle metriche per la ricostruzione dei volumi con interpolazione naive

Metrica	Mean	Std	Min	Max	Median
MSE	$3.14 \cdot 10^{-5}$	$8.59 \cdot 10^{-6}$	$2.13 \cdot 10^{-5}$	$5.07 \cdot 10^{-5}$	$2.97 \cdot 10^{-5}$
MAE	0.0037	0.0006	0.0030	0.0049	0.0036
RMSE	0.0056	0.0007	0.0046	0.0071	0.0054
PSNR	45.1790	1.1343	42.9516	46.7160	45.2739
SSIM	0.9834	0.0034	0.9760	0.9876	0.9841
NCC	0.9995	0.0001	0.9993	0.9996	0.9995
SNR	30.2805	0.8409	28.8394	31.8512	30.2090

(b) Risultati delle metriche per la ricostruzione dei volumi con il modello a 2 angoli

Metrica	Mean	Std	Min	Max	Median
MSE	$3.39 \cdot 10^{-5}$	$9.84 \cdot 10^{-6}$	$2.29 \cdot 10^{-5}$	$5.70 \cdot 10^{-5}$	$3.15 \cdot 10^{-5}$
MAE	0.0038	0.0006	0.0031	0.0051	0.0037
RMSE	0.0058	0.0008	0.0048	0.0076	0.0056
PSNR	44.8676	1.1772	42.4391	46.4022	45.0228
SSIM	0.9824	0.0038	0.9736	0.9868	0.9833
NCC	0.9994	0.0001	0.9992	0.9996	0.9994
SNR	29.9690	0.8712	28.5070	31.5375	29.9201

(c) Risultati delle metriche per la ricostruzione dei volumi con il modello a 4 angoli

Tabella 4.3: Risultati delle slice ricostruite tramite tecniche, rispetto alle slice ricostruite con 240 proiezioni originali

Attraverso i risultati ottenuti, si confermano significativi miglioramenti per i modelli, osservabili in tutte le metriche considerate.

Metrica	Interp. Lineare	2 angoli	Miglioramento (%)
MSE	$4.18 \cdot 10^{-4}$	$3.14 \cdot 10^{-5}$	+92.5%
MAE	0.0118	0.0037	+68.3%
RMSE	0.0203	0.0056	+72.6%
PSNR	33.9417	45.1790	+33.1%
SSIM	0.8578	0.9834	+14.6%
NCC	0.9928	0.9995	+0.7%
SNR	19.0432	30.2805	+59.0%

(a) Confronto metriche medie in ricostruzione fra interpolazione naive e modello a 2 angoli

Metrica	Interp. Lineare	4 angoli	Miglioramento (%)
MSE	$4.18 \cdot 10^{-4}$	$3.39 \cdot 10^{-5}$	+91.9%
MAE	0.0118	0.0038	+67.3%
RMSE	0.0203	0.0058	+71.6%
PSNR	33.9417	44.8676	+32.2%
SSIM	0.8578	0.9824	+14.5%
NCC	0.9928	0.9994	+0.7%
SNR	19.0432	29.9690	+57.4%

(b) Confronto metriche medie in ricostruzione fra interpolazione naive e modello a 4 angoli

Tabella 4.4: Confronti sulle metriche medie delle slice

Tramite le Tabelle 4.4a e 4.4b, è possibile notare come la distribuzione dei miglioramenti differisca rispetto a quella ottenuta durante la comparazione fra proiezioni. La motivazione è principalmente da ricercare nella natura del confronto: infatti, comparando le proiezioni, è possibile osservare direttamente le differenze prodotte dalle tecniche, senza introdurre variazioni dovute alla ricostruzione.

Nonostante ciò, le analisi mostrano ugualmente un miglioramento comparabile fra i due modelli, sebbene i miglioramenti ottenuti risultino leggermente ridotti per il modello a 4 angoli di visione. Per questo motivo è stato considerato standard il modello con 2 angoli di visione.

4.2 Esperimenti da 120 a 480 viste

Osservando i miglioramenti ottenuti tramite i precedenti incrementi di proiezioni, si è sperimentata un'ulteriore applicazione delle reti realizzate, passando da 240 proiezioni a 480.

4.2.1 Riallenamento e dataset

Inizialmente per questa analisi si sono impiegati unicamente i modelli già presentati nei capitoli precedenti. Si sono quindi utilizzati come input per i modelli gli output della precedente applicazione, realizzando di fatto un incremento da 120 a 480 proiezioni interamente tramite reti neurali. Per motivi di semplicità sono stati utilizzati come input ai modelli, i loro relativi output dell'applicazione precedente.

Secondariamente si è sperimentato un riallenamento dei due modelli per introdurre una maggiore specializzazione nell'interpolazione a 480 proiezioni. A questo scopo è stato generato un nuovo dataset, fissando la distanza angolare a 0.75 gradi, l'intervallo angolare a $[0, 360]$ gradi e il numero di proiezioni a $N = 480$ per ogni volume.

Si riportano in Figura 4.6 e in Figura 4.7 i grafici di riallenamento dei modelli.

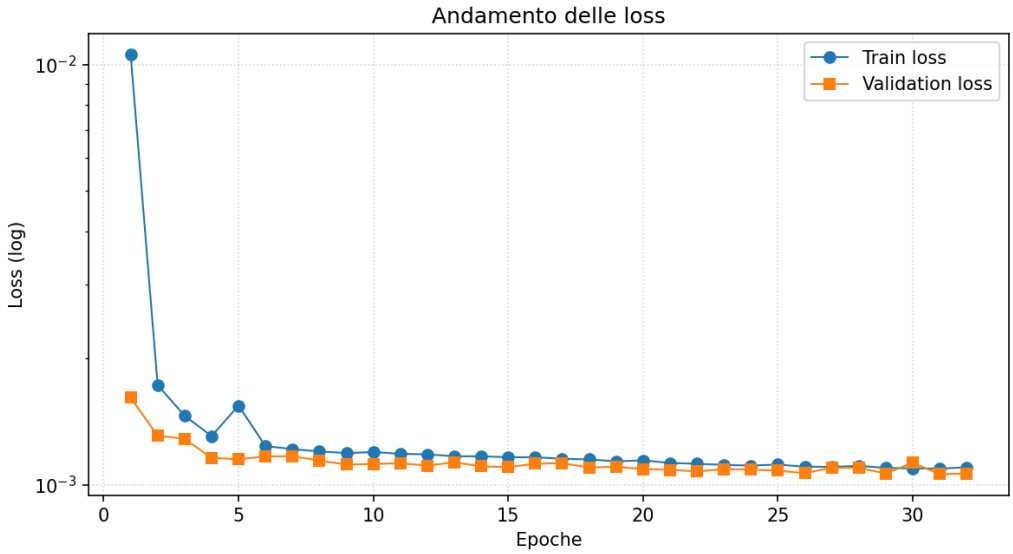


Figura 4.6: Andamento delle loss del modello con 2 angoli riallenato

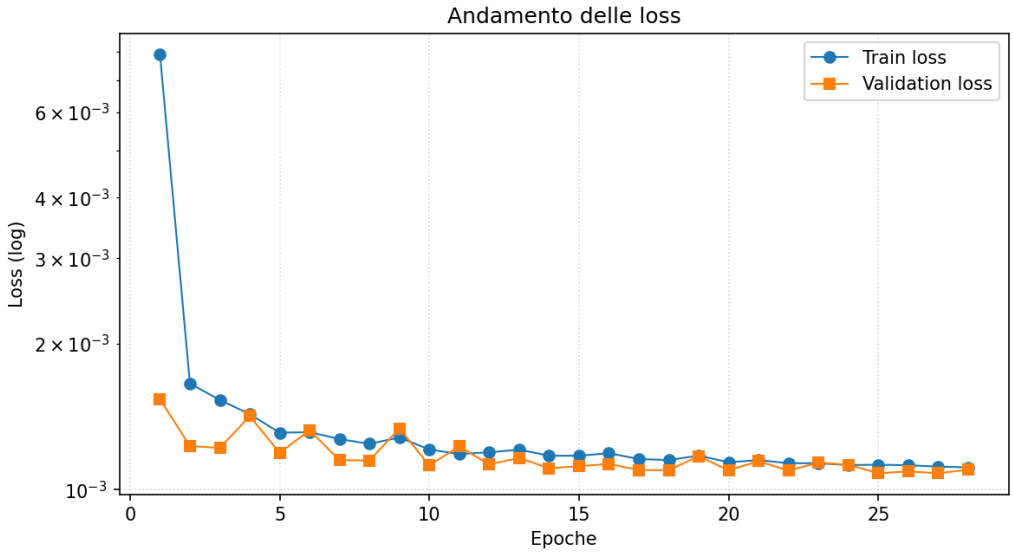
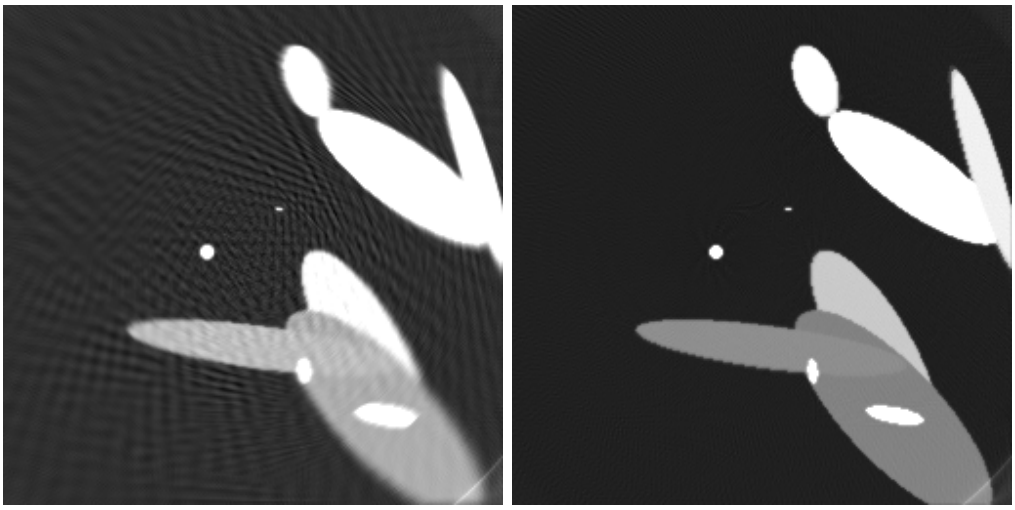


Figura 4.7: Andamento delle loss del modello con 4 angoli riallenato

4.2.2 Confronti

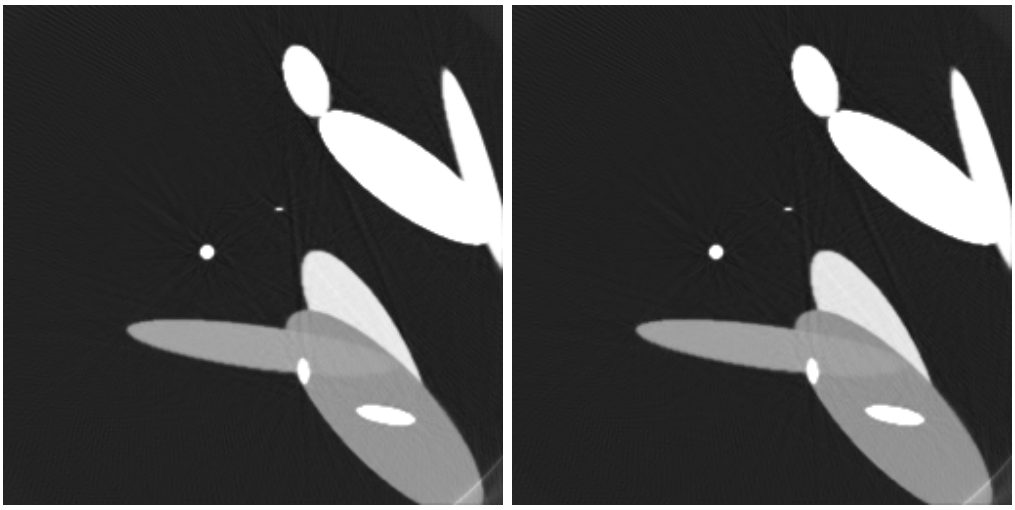
Per maggiore chiarezza nei confronti, si riportano le slice dei volumi ricostruiti tramite proiezioni interpolate con interpolazione lineare e tramite 480 proiezioni originali (Figura 4.8).



(a) Interpolazione lineare (b) Originale 480 angoli

Figura 4.8: Confronto fra volumi ricostruiti con proiezioni interpolate e con proiezioni originali

Considerando la ridotta qualità della ricostruzione effettuata tramite interpolazione lineare, essa è stata esclusa dalle seguenti valutazioni. Si riporta in Figura 4.9 un confronto fra slice ottenute tramite il modello a 2 angoli di visione senza e con riallenamento.



(a) Senza riallenamento (b) Con riallenamento

Figura 4.9: Confronto fra slice ottenute dal modello a 2 angoli, senza e con riallenamento

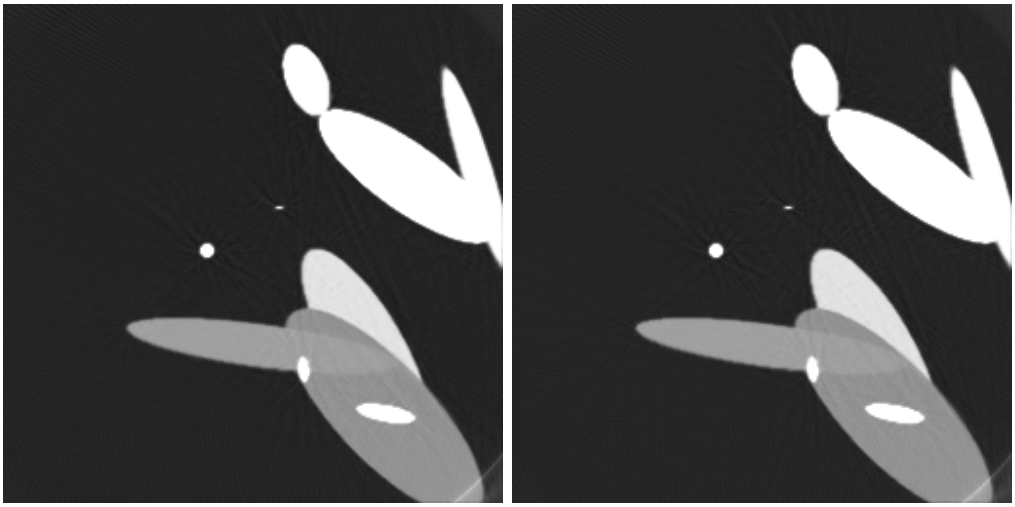
Si riportano nella Tabella 4.5 le relative metriche di media e deviazione standard sulla base di volumi ricostruiti con 480 proiezioni originali, confrontando l'effetto del riallenamento sui modelli.

Metrica	**Senza riallenamento**		**Con riallenamento**	
	Mean	**Std**	**Mean**	**Std**
MSE	$3.10 \cdot 10^{-5}$	$8.40 \cdot 10^{-6}$	$2.79 \cdot 10^{-5}$	$7.96 \cdot 10^{-6}$
MAE	0.0031	0.0005	0.0030	0.0005
RMSE	0.0055	0.0007	0.0052	0.0007
PSNR	45.2397	1.1216	45.7058	1.1911
SSIM	0.9876	0.0028	0.9884	0.0027
NCC	0.9995	0.0001	0.9995	0.0001
SNR	30.3269	0.8092	30.7930	0.8390

Tabella 4.5: Confronti sulle metriche delle slice del modello a 2 angoli senza e con riallenamento

In questa prima fase è possibile notare un miglioramento da parte del

modello riallenato. Tuttavia, esso risulta essere contenuto rispetto al risultato del modello di base. Si ripropone un confronto fra slice per il modello con 4 angoli di visione, senza e con riallenamento, in Figura 4.10.



(a) Senza riallenamento (b) Con riallenamento

Figura 4.10: Confronto fra slice ottenute dal modello a 4 angoli, senza e con riallenamento

Si riportano nella Tabella 4.6 i relativi risultati e confronti.

Metrica	**Senza riallenamento**		**Con riallenamento**	
	Mean	**Std**	**Mean**	**Std**
MSE	$3.17 \cdot 10^{-5}$	$8.92 \cdot 10^{-6}$	$3.11 \cdot 10^{-5}$	$9.14 \cdot 10^{-6}$
MAE	0.0031	0.0005	0.0031	0.0005
RMSE	0.0056	0.0008	0.0055	0.0008
PSNR	45.1457	1.1486	45.2439	1.2064
SSIM	0.9876	0.0029	0.9875	0.0030
NCC	0.9995	0.0001	0.9995	0.0001
SNR	30.2329	0.8297	30.3311	0.8784

Tabella 4.6: Confronti sulle metriche delle slice del modello a 4 angoli senza e con riallenamento

4.2.3 Analisi sui risultati

Come evidenziato dai confronti grafici e numerici, l'incremento da 240 a 480 proiezioni risulta proficuo per il miglioramento delle ricostruzioni, riducendo gli artefatti grafici dei volumi ricostruiti con 120 e 240 proiezioni (riportati in Figura 4.4).

È anche possibile notare come l'incremento derivato dal riallenamento della rete migliori parzialmente i risultati ottenuti. Nonostante ciò, è fondamentale notare come questo riallenamento richieda una notevole quantità di stack di proiezioni da 480 per un corretto allenamento. In numerose applicazioni medicali, non sono disponibili stack di proiezioni di questa dimensione, di conseguenza può risultare un ostacolo in determinati ambienti, rispetto ad una quantità di proiezioni ridotta della metà. Tuttavia risulta una opzione percorribile in caso di abbondante disponibilità di dati.

Anche in caso di scarsa disponibilità però, possiamo notare come anche il modello allenato su 240 proiezioni possa portare alla generazione di volumi di ottima qualità, indicando una complessivamente buona generalizzazione del problema.

Conclusioni

In questa tesi si sono sviluppate tecniche di deep learning per il miglioramento delle ricostruzioni tomografiche, attraverso l'aumento del numero di proiezioni. In particolare, sono state proposte diverse varianti che si differenziano per la quantità di proiezioni in ingresso e le strategie di gestione dei dati, tra le quali il modello basato su 2 proiezioni è stato considerato lo standard.

Per valutare l'efficacia dell'utilizzo del deep learning, sono stati condotti confronti con tecniche di interpolazione classiche, utilizzando metriche di valutazione della qualità delle immagini. Le metriche raccolte hanno evidenziato come l'approccio tramite reti neurali possa aumentare significativamente la fedeltà di generazione delle proiezioni, permettendo significativi miglioramenti alla qualità dei volumi finali ricostruiti.

Nello specifico, sono state calcolate le metriche su due principali casi di studio: l'aumento da 120 a 240 proiezioni e quello da 120 a 480 proiezioni. Nel primo caso, considerando esclusivamente il modello standard, il volume ricostruito a partire da 240 proiezioni (di cui la metà generate artificialmente dalla rete neurale) risulta visibilmente più nitido e fedele rispetto a quello ottenuto con le sole 120 proiezioni originali, sebbene permangano lievi artefatti. Nel secondo esperimento, la stessa rete, inizialmente addestrata per passare da 120 a 240 proiezioni, è stata nuovamente applicata alle 240 proiezioni precedenti, portando così a un totale di 480 proiezioni (di cui tre quarti sono sintetiche). La ricostruzione ottenuta in questo caso mostra ancora un'elevata qualità visiva e una riduzione ulteriore degli artefatti, a fronte

di un incremento del costo computazionale dell'algoritmo FDK comunque contenuto.

Per affinare il passaggio da 240 a 480 proiezioni è infine stato testato un riallenamento specifico delle reti, modificando il dataset di allenamento. Questo processo ha portato benefici marginali alle ricostruzioni finali e presuppone la disponibilità di un numero molto elevato di stack di proiezioni per l'addestramento (una condizione spesso irrealistica in contesti medici reali). Pertanto, se da un lato il riallenamento non è risultato particolarmente vantaggioso, dall'altro l'utilizzo sequenziale della rete originaria si è dimostrato un approccio efficace e robusto, permettendo una adeguata generalizzazione del problema anche per ulteriori applicazioni successive.

In conclusione, i risultati sperimentali mostrano che la configurazione proposta in questa tesi consente di ottenere ricostruzioni 3D con un livello di dettaglio elevato e una riduzione significativa degli artefatti, rispetto ai metodi tradizionali, senza introdurre costi computazionali rilevanti in fase di utilizzo clinico. Tali risultati evidenziano il potenziale delle reti neurali come strumento complementare alle tecniche tomografiche moderne, con prospettive concrete di applicazione in contesti clinici reali.

Bibliografia

[1] Jiang Hsieh. Computed tomography: principles, design, artifacts, and recent advances. 2003.

[2] Chun-Feng Cao, Kun-Long Ma, Hua Shan, Tang-Fen Liu, Si-Qiao Zhao, Yi Wan, and Hai-Qiang Wang. Ct scans and cancer risks: a systematic review and dose-response meta-analysis. *BMC cancer*, 22(1):1238, 2022.

[3] Lawrence A Shepp and Benjamin F Logan. The fourier reconstruction of a head section. *IEEE Transactions on nuclear science*, 21(3):21–43, 1974.

[4] Lee A Feldkamp, Lloyd C Davis, and James W Kress. Practical cone-beam algorithm. *Journal of the Optical Society of America A*, 1(6):612–619, 1984.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[7] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[9] Furkat Safarov, Kuchkorov Temurbek, Djumanov Jamoljon, Ochilov Temur, Jean Chamberlain Chedjou, Akmalbek Bobomirzaevich Abdusalomov, and Young-Im Cho. Improved agricultural field segmentation in satellite imagery using tl-resunet architecture. *Sensors*, 22(24):9784, 2022.

[10] Sahil Gangurde. Building and road segmentation using effunet and transfer learning approach. *arXiv preprint arXiv:2307.03980*, 2023.

[11] Carolyn Christiansen and Gengsheng L Zeng. Sinogram interpolation inspired by single-image super resolution. *Journal of biotechnology and its applications*, 2(1):1010, 2023.

[12] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4331–4339, 2019.

[13] Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on image processing*, 6(2):298–311, 1997.

[14] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The thrity-seventh asilomar conference on signals, systems & computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[15] Wim Van Aarle, Willem Jan Palenstijn, Jan De Beenhouwer, Thomas Altantzis, Sara Bals, K Joost Batenburg, and Jan Sijbers. The astra toolbox: A platform for advanced algorithm development in electron tomography. *Ultramicroscopy*, 157:35–47, 2015.

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Ringraziamenti

Voglio inizialmente ringraziare la mia relatrice Elena Morotti e la mia correlatrice Elena Loli Piccolomini per il supporto dedicatomi durante la stesura di questa tesi.

Porgo un ringraziamento speciale alla mia famiglia, che mi ha supportato in questi anni di studio, mostrandomi vicinanza e abbattendo ostacoli che sarebbero stati insormontabili altrimenti.

Dedico inoltre un ringraziamento speciale ai miei compagni di studi, Davide, Elia, Flavio, Francesco, Leonardo e Lorenzo, i quali mi hanno accompagnato anche nei periodi più duri. Con voi ho certamente condiviso periodi di studio, ma soprattutto ho vissuto momenti di crescita, risate inattese e tantissime belle serate. Sono contento di aver condiviso con voi eventi, viaggi e ricordi meravigliosi.

Ringrazio inoltre tutti i miei amici più stretti, del gruppo di "Puglia", con i quali non ho condiviso libri, ma notti infinite. Con voi ogni notte erano chiacchiere, ricordi, opinioni, risate, confidenze, canzoni, progetti impossibili e molte altre cose. Sono contento di aver vissuto tutto questo e di essere cresciuto insieme a voi.

Porgo infine un ringraziamento a tutte quelle persone che mi hanno aiutato nel mio percorso, anche se non presenti in queste poche righe.