ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea in Informatica

Tomografia traslazionale 3D per lo screening bagagli con riduzione degli artefatti tramite Residual U-Net

Relatore:
Chiar.ma Prof.
Elena Loli Piccolomini

Correlatore:
Chiar.mo Dott.
Davide Evangelista

Presentata da:
Simone Ballo

Sessione II
Anno Accademico 2024/2025

Introduzione

La tomografia computerizzata è una delle tecniche di imaging più diffuse ed efficaci per l'analisi non invasiva e la ricostruzione tridimensionale di oggetti e volumi. Sebbene il suo impiego sia nato e si sia consolidato principalmente in ambito medico, negli ultimi anni essa ha trovato applicazione anche nel settore della sicurezza aeroportuale, dove si rivela uno strumento fondamentale per lo screening dei bagagli. L'utilizzo di scanner CT non velocizza i controlli rispetto ai tradizionali sistemi X-ray 2D, ma offre un vantaggio decisivo in termini di accuratezza e affidabilità, reso possibile dalla ricostruzione tridimensionale del contenuto dei bagagli, che permette di individuare con maggiore efficacia gli oggetti potenzialmente pericolosi e di ridurre, di conseguenza, il numero di ispezioni manuali necessarie.

In questo contesto si inserisce la tomografia traslazionale, una variante della tomografia tradizionale che, grazie al movimento lineare della sorgente e del detector, si adatta perfettamente alla configurazione tipica degli aeroporti, dove i bagagli scorrono su un nastro trasportatore.

Il presente lavoro di tesi si propone di simulare questo scenario tramite l'impiego della libreria ASTRA Toolbox, sviluppando tecniche di ricostruzione specifiche per il caso descritto. A tal fine è stata realizzata una geometria *custom*, capace di riprodurre in maniera fedele il funzionamento di uno scanner CT durante lo screening dei bagagli. Successivamente è stato sperimentato l'algoritmo algebrico di ricostruzione SIRT (Simultaneous Iterative Reconstruction Technique) che, affiancato a un modello di Residual U-Net progettato per ridurre gli artefatti causati dalla limitata quantità di proiezioni, ha generato ricostruzioni qualitativamente valide.

La struttura della tesi è organizzata come segue:

- Il Capitolo 1 introduce i principi teorici della tomografia, analizzando i concetti di proiezione, il teorema della sezione di Fourier e la retroproiezione, per poi approfondire il contesto applicativo della sicurezza aeroportuale e le peculiarità della tomografia traslazionale.

- Il Capitolo 2 presenta gli strumenti e le metodologie adottate: la libreria ASTRA Toolbox per la simulazione, gli algoritmi iterativi di ricostruzione, con particolare attenzione al metodo SIRT, e i principi teorici delle reti neurali convoluzionali, base per la progettazione del modello di Residual U-Net.
- Il Capitolo 3 descrive il caso di studio, partendo dal processo di generazione del dataset sintetico utilizzato per le simulazioni. Vengono quindi introdotte le principali geometrie rotazionali già presenti in ASTRA, per poi illustrare nel dettaglio la realizzazione della geometria traslazionale *custom*, sviluppata appositamente per riprodurre lo scenario dello screening aeroportuale. Infine, si presentano le caratteristiche e il processo di addestramento del modello di Residual U-Net impiegato per la rimozione degli artefatti dalle ricostruzioni ottenute.
- Il Capitolo 4 presenta e discute i risultati ottenuti, valutando sia l'accuratezza della ricostruzione tramite ASTRA, sia l'efficacia del modello di deep learning nell'incrementare la qualità dei volumi ricostruiti.

Indice

Elenco delle tabelle

Capitolo 1

Tomografia computerizzata

Introdotta alla fine degli anni '70, la **tomografia computerizzata (CT)** ha rivoluzionato le tecniche di imaging grazie alla possibilità di ottenere rappresentazioni tridimensionali degli oggetti senza interventi invasivi. La sua efficacia si basa sull'acquisizione di proiezioni generate dall'attenuazione dei raggi X e sulla loro successiva elaborazione mediante algoritmi di ricostruzione, che permettono di analizzare in dettaglio la struttura interna del volume.

Per acquisire le proiezioni necessarie alla ricostruzione sono indispensabili due componenti principali: la **sorgente** di raggi X, che emette il fascio radiante, e il **detector**, che misura l'intensità residua dopo l'attraversamento dell'oggetto. La configurazione più diffusa è quella **rotazionale**, in cui sorgente e detector compiono un movimento circolare attorno al volume in esame. In questo modo vengono raccolte proiezioni da angolazioni differenti, che complessivamente forniscono le informazioni necessarie a ricostruire in maniera accurata la struttura interna nelle tre dimensioni.

In questo capitolo vengono presentati i fondamenti teorici su cui poggia la tomografia: dal concetto di proiezione e integrali di linea, al **Fourier-Slice Theorem**, fino alla tecnica di retroproiezione. Successivamente, si analizza l'ambito di applicazione di questa tesi e si introduce a livello teorico la **tomografia traslazionale**. L'obiettivo è fornire un inquadramento teorico utile a comprendere il funzionamento degli algoritmi di ricostruzione e le motivazioni dietro le scelte implementative adottate in questo lavoro.

1.1 Proiezione

Uno dei concetti fondamentali alla base della tomografia è quello dell'**integrale di linea**. Esso consiste nell'integrare una funzione che viene valutata lungo un cammino o una retta. Nel caso della tomografia a raggi X, l'oggetto viene descritto come una distribuzione bidimensionale (o tridimensionale) del coefficiente di attenuazione dei raggi, indicato con $\mu(x, y)$.

L'intensità del fascio che attraversa l'oggetto decresce secondo la **legge di Beer-Lambert**:

$$I = I_0 \, e^{-\int_{\text{linea}} \mu(x,y)\, ds} \tag{1.1}$$

dove I_0 è l'intensità iniziale del fascio. In pratica, l'integrale di linea corrisponde alla perdita totale di intensità che un fascio di raggi X subisce quando attraversa l'oggetto lungo quella direzione.

Per descrivere questo processo matematico, si introduce la funzione $f(x, y)$ che rappresenta il coefficiente di attenuazione dei raggi X dell'oggetto in due dimensioni (ossia $\mu(x, y)$). L'integrale di linea è descritto dalla coppia (θ, t), dove θ rappresenta l'angolo di orientamento della linea e t rappresenta la distanza dell'origine dal raggio. Dunque, l'equazione della retta è:

$$x \cos\theta + y \sin\theta = t \tag{1.2}$$

La proiezione $P_\theta(t)$ è definita come l'integrale della funzione $f(x, y)$ lungo questa retta. Sfruttando la **funzione delta di Dirac** possiamo riscrivere $P_\theta(t)$ come un doppio integrale vincolato alla retta definita da (θ, t):

$$P_\theta(t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)\, \delta(x \cos\theta + y \sin\theta - t)\, dx\, dy \tag{1.3}$$

Questa funzione prende il nome di **trasformata di Radon** della funzione $f(x, y)$ e si utilizza per descrivere la proiezione di un oggetto. Una proiezione non è altro che la combinazione di un insieme di integrali di linea. Quella più semplice è quella parallela (Fig. 1.1), ottenuta muovendo sorgente e detector in direzioni parallele sui lati opposti dell'oggetto[1].

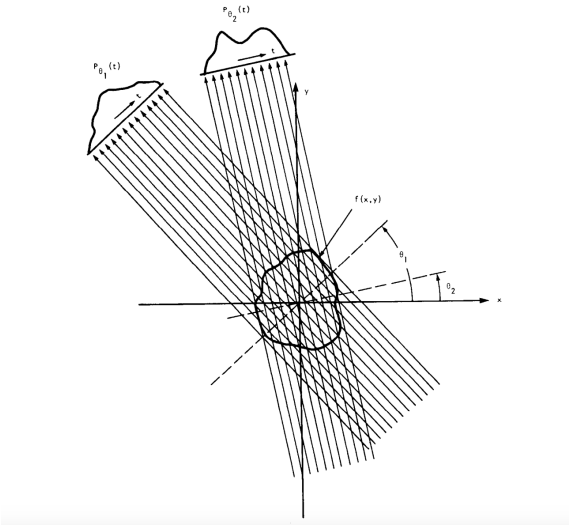


Figura 1.1: Proiezione parallela $P_\theta(t)$ della figura

Un altro tipo di acquisizione è la **fan-beam**, che utilizza una sorgente fissa e dispone i raggi non in parallelo, ma bensì 'a ventaglio' (Fig. 1.2).

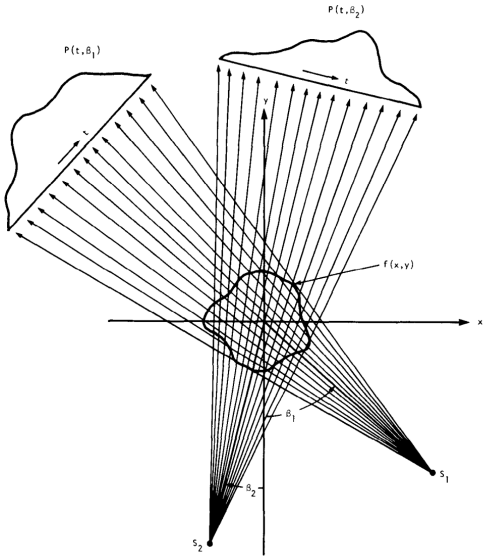


Figura 1.2: Proiezione fan-beam sulla figura $f(x, y)$

Si introduce il *Fourier-slice Theorem*, fondamentale per comprendere il perché sia possibile ricostruire un oggetto a partire dalle sue proiezioni.

1.2 Fourier-slice Theorem

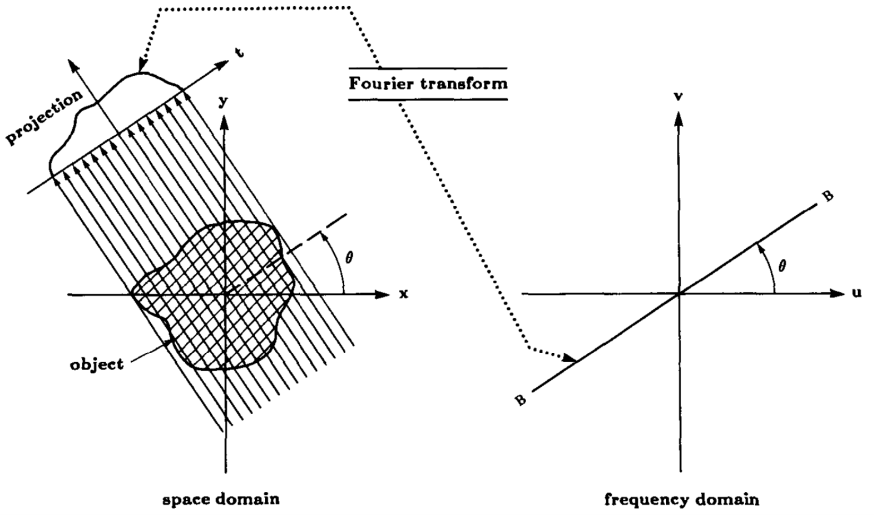


Figura 1.3: Il teorema della sezione di Fourier mette in relazione la trasformata di Fourier di una proiezione con la trasformata di Fourier dell'oggetto lungo una linea radiale.

Il Fourier-slice Theorem[1] fornisce una relazione tra le proiezioni di una funzione e la funzione stessa; esso afferma che la trasformata di Fourier della proiezione di f su un angolo θ corrisponde a una sezione della trasformata di Fourier bidimensionale dell'oggetto lungo una retta passante per l'origine, inclinata dello stesso angolo. (Fig. 1.3).

In sintesi, ogni proiezione (ottenuta tramite la trasformata di Radon) fornisce una 'striscia' di informazioni sul contenuto in frequenza dell'oggetto. Raccogliendo proiezioni da più angoli diversi, è possibile coprire l'intero piano delle frequenze e, una volta noto ciò, l'oggetto può essere ricostruito applicando la trasformata di Fourier inversa[2].

In questo contesto, per 'frequenza' si intende quanto rapidamente varia l'intensità dell'oggetto nello spazio: basse frequenze corrispondono a variazioni lente e strutture grandi, mentre alte frequenze rappresentano dettagli fini e contorni.

Questo teorema costituisce la base di metodi di ricostruzione analitici come *Filtered BackProjection (FBP).*

1.3 Retroproiezione

La retroproiezione è un metodo utilizzato per risolvere il problema inverso nella tomografia computerizzata e consiste nel ricostruire la distribuzione spaziale del coefficiente di attenuazione $\mu(x, y)$, a partire da un numero finito di dati di proiezione raccolti da diverse angolazioni. Sommando i contributi di ciascuna immagine retroproiettata su un'unica sezione, si ottiene la ricostruzione tomografica dell'oggetto. (Fig. 1.4).

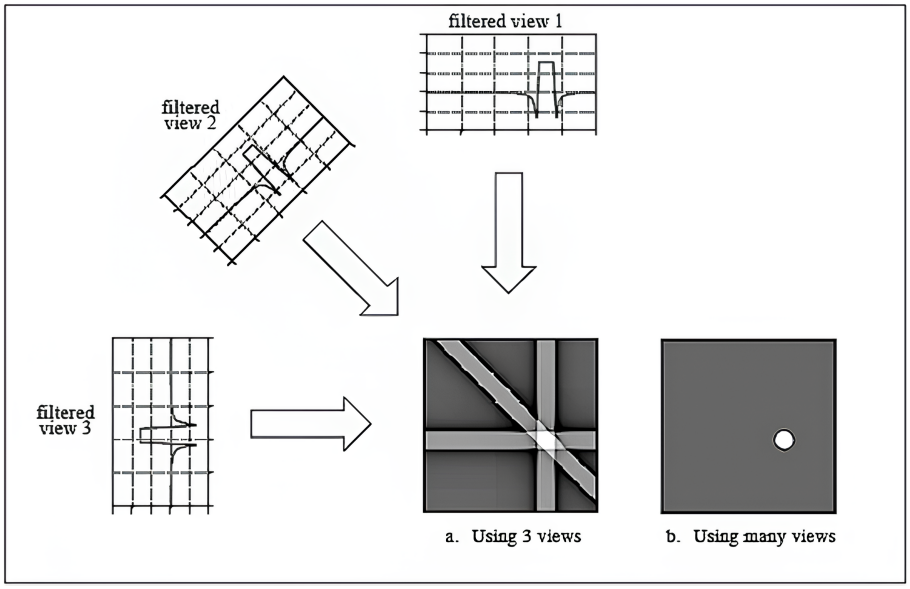


Figura 1.4: Le tre proiezioni raccolte vengono proiettate all'indietro per ricostruire l'immagine originale

Maggiore è il numero di viste/proiezioni a disposizione, maggiore sarà la qualità dell'oggetto ricostruito.

1.4 Tomografia in ambito aeroportuale

La tomografia computerizzata riveste un ruolo sempre più importante nella sicurezza aeroportuale. Oggi la CT viene impiegata nell'ispezione di bagagli a mano e da stiva, permettendo di rilevare oggetti nascosti o potenzialmente pericolosi con maggiore affidabilità e precisione rispetto ai tradizionali sistemi a raggi X bidimensionali. I progressi tecnologici degli ultimi anni hanno reso gli scanner più compatti e facilmente integrabili nei sistemi di sicurezza, consentendo la scansione di grandi volumi di bagagli senza rallentare il flusso di passeggeri.

Le procedure di sicurezza aeroportuali si sono evolute nel tempo, in particolare in seguito a eventi come gli attentati dell'11 settembre 2001, che hanno reso necessari controlli più lunghi e rigorosi. Le tecnologie CT moderne, integrate con tecniche avanzate come la spettroscopia molecolare Raman[3], permettono di identificare sostanze pericolose e dispositivi elettronici all'interno di contenitori chiusi, riducendo la necessità di controlli manuali e migliorando conseguentemente l'esperienza complessiva dei viaggiatori[4].

Tuttavia, il cambiamento procede lentamente. Nel 2022, il Regno Unito ha fissato la scadenza di giugno 2024 per l'adozione di questa tecnologia in tutti gli aeroporti principali, ma molti di questi non sono riusciti a rispettare tale scadenza, richiedendo delle proroghe. L'ostacolo principale risiede nel prezzo per singolo scanner che può variare da 80.000 a 300.000 USD. Aziende terze stanno iniziando a fornire soluzioni personalizzate per integrare la CT nei processi di ispezione, offrendo spesso alternative a costi ridotti.

In futuro, grazie a un maggior utilizzo di questa tecnologia e a una maggiore automazione, i tempi di attesa all'interno dell'aeroporto potrebbero diminuire drasticamente e potrebbe non essere più necessario arrivare ore prima del volo.

Poiché nei controlli aeroportuali non è possibile ruotare i bagagli durante l'ispezione, si impiega una variante della tomografia rotazionale, nota come **tomografia traslazionale** (vedi sezione successiva).

1.5 Tomografia traslazionale

La tomografia traslazionale, a differenza di quella rotazionale — che prevede una rotazione attorno all'oggetto — si basa su un movimento lineare (traslazione). Sorgente e detector si muovono in parallelo lungo lo stesso asse e acquisiscono proiezioni da diverse posizioni lineari anziché da diverse angolazioni.

La principale differenza risiede nella modalità di acquisizione dei dati. Nella tomografia rotazionale, la rotazione attorno all'oggetto consente di raccogliere in modo naturale informazioni su tutte le dimensioni del volume, mentre nella tomografia traslazionale è necessario disporre più coppie di sorgenti e detector su diverse linee parallele per ottenere una copertura equivalente. Nonostante questa limitazione, la semplicità meccanica offerta da un approccio di questo tipo lo rende spesso preferibile. In molte applicazioni, muovere o ruotare l'oggetto risulta infatti impraticabile o svantaggioso; in questi casi, è più semplice far muovere sorgente e detector lungo una traiettoria lineare.

Capitolo 2

Strumenti e metodologie

Dopo aver introdotto i principi teorici alla base della tomografia, questo capitolo descrive gli strumenti e le metodologie utilizzate per lo sviluppo del lavoro. In particolare, si presenta la libreria **ASTRA Toolbox**, specializzata in simulazioni tomografiche.

Successivamente, viene approfondito il funzionamento degli algoritmi iterativi di ricostruzione algebrica con particolare attenzione ai metodi **ART** e **SIRT** . Infine, viene illustrata la teoria alla base delle reti neurali convoluzionali e descritto il funzionamento della **Residual U-Net**, un'architettura di deep learning fondamentale nella rimozione degli artefatti di ricostruzione.

2.1 ASTRA Toolbox

ASTRA toolbox è una libreria open-source sviluppata in C++ con interfacce in Python e MATLAB, progettata per applicazioni di tomografia computazionale. Grazie alla sua versatilità, questa libreria si presta efficacemente alla simulazione tomografica in diversi contesti applicativi.

Infatti, oltre a fornire supporto per diverse geometrie di acquisizione, sia bidimensionali che tridimensionali (come *parallel*, *fan-beam* e *cone-beam*), ASTRA consente anche di definire geometrie personalizzate, permettendo di modellare scenari sperimentali specifici.

Da un punto di vista algoritmico, la libreria mette a disposizione sia metodi analitici (come FBP), sia metodi iterativi (come SIRT, SART e CGLS) per ricostruire l'oggetto a partire dalle sue proiezioni.

Un altro punto di forza di questo strumento risiede nello sfruttamento dell'accelerazione hardware della GPU tramite CUDA.

CUDA è un'architettura hardware per l'elaborazione parallela, sviluppata da NVIDIA, che permette di sfruttare la GPU non solo per elaborazioni grafiche, ma anche per eseguire calcoli generici. Grazie a questa tecnologia, ASTRA è in grado di simulare e ricostruire oggetti tridimensionali riducendo significativamente i tempi di calcolo rispetto a un'elaborazione basata puramente su CPU.

2.2 Algoritmi algebrici

Gli approcci classici, basati sulla trasformata di Radon e sulla trasformata di Fourier, funzionano bene solo quando le proiezioni raccolte sono numerose e distribuite in modo regolare lungo un angolo completo di 180° o 360°. Questo è il caso della tomografia medica, come la TAC, dove il macchinario ruota attorno al paziente raccogliendo proiezioni uniformi. In altri contesti applicativi — come quello aeroportuale — non è tuttavia possibile disporre di una quantità di dati così ampia e uniformemente distribuita. In questi scenari, gli algoritmi basati sulle trasformate perdono di efficacia e precisione; di conseguenza, si ricorre a un approccio alternativo noto come **metodo algebrico**.

L'idea è quella di suddividere l'immagine da ricostruire in una **griglia di celle** (o voxel, nel caso di volumi 3D), assumendo che all'interno di ciascuna cella il valore dei *pixel* sia costante. Quando un raggio attraversa l'immagine, esso passa all'interno di alcune celle della griglia e fornisce in output un valore chiamato **ray-sum**. Esso non è altro che la somma dei valori delle celle attraversate, pesate in base alla distanza percorsa dal raggio all'interno di ciascuna cella.
Ogni *ray-sum* può quindi essere interpretato come un'equazione lineare:

$$\sum_{j=1}^{N} W_{ij} f_j = p_i \tag{2.1}$$

dove f_j è il valore sconosciuto da determinare della cella j, p_i è il *ray-sum* misurato dal raggio i, e W_{ij} è il peso che indica quanto il raggio i ha attraversato la cella j.

Raggruppando tutte le equazioni dei raggi provenienti da tutte le proiezioni si ottiene un sistema di equazioni lineari, le cui incognite sono i valori delle celle dell'immagine[1]. Nei casi reali, un sistema di questo tipo può essere composto da decine di migliaia di equazioni con altrettante incognite.

Per comprendere al meglio come funzionano questi tipi di algoritmi, si introduce il concetto di **sinogramma**.

Il sinogramma è una rappresentazione grafica delle proiezioni acquisite dell'immagine (Fig. 2.1). Ogni riga del sinogramma corrisponde a una proiezione a un certo angolo, mentre ogni colonna corrisponde a un raggio particolare in quella proiezione. In altre parole, la posizione (i, j) contiene il *ray-sum* fornito dal j-esimo raggio della i-esima proiezione. Questa rappresentazione racchiude, dunque, i dati misurati necessari per costruire e risolvere il sistema di equazioni lineari, che a sua volta permette di ricostruire l'immagine originale.

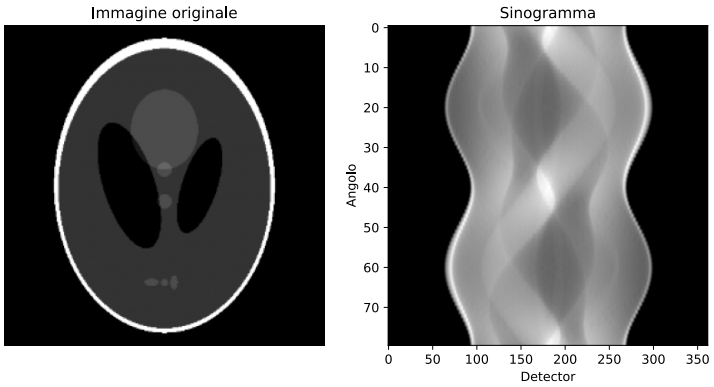


Figura 2.1: A sinistra è riportata l'immagine originale, mentre a destra è mostrato il sinogramma ottenuto dalle proiezioni dell'immagine.

Introdotto questo concetto, possiamo definire più nel dettaglio il funzionamento dell'algoritmo descritto dal **metodo di Kaczmarz**.
Il procedimento iterativo utilizzato parte da una **stima iniziale** dei valori dei *pixel*, indicata con F_0, che in genere consiste in un'immagine composta da soli zeri. A partire da F_0, si calcolano i **ray-sum simulati** utilizzando la stessa geometria e disposizione delle proiezioni originali.

Per ciascun raggio, si confronta il *ray-sum* simulato con quello misurato nel sinogramma reale. La differenza tra questi due valori indica di quanto i *pixel* attraversati dal raggio debbano essere corretti. Tale correzione viene distribuita tra i *pixel* in proporzione ai loro pesi nell'equazione associata al raggio: quelli maggiormente attraversati subiscono correzioni più consistenti rispetto a quelli attraversati in misura minore. Ripetendo questo processo per tutti i raggi e per tutte le proiezioni, si ottiene una nuova stima dell'immagine.

La formula di aggiornamento utilizzata è dunque la seguente:

$$f^{(k+1)} = f^{(k)} + \frac{p_i - \sum_{j=1}^{N} W_{ij} f_j^{(k)}}{\sum_{j=1}^{N} W_{ij}^2} W_i \tag{2.2}$$

Dove:

- p_i è il *ray-sum* reale misurato dal raggio i.
- W_{ij} è il peso che indica quanto il raggio i ha attraversato la cella j.
- $W_i = (W_{i1}, W_{i2}, \ldots, W_{iN})$ è il vettore dei pesi del raggio i.
- $\sum_{j=1}^{N} W_{ij} f_j^{(k)}$ è il *ray-sum* stimato.
- $\sum_{j=1}^{N} W_{ij}^2$ normalizza l'aggiornamento in base alla lunghezza di W_i

L'algoritmo procede in modo iterativo, aggiornando la stima ad ogni ciclo, fino a quando non vengono prodotte delle proiezioni che si avvicinino sufficientemente a quelle originali.

Se esiste un'unica rappresentazione compatibile con tutte le proiezioni, questo procedimento converge alla soluzione corretta. In presenza di rumore o di sistemi sottodeterminati, in cui il numero di equazioni è minore del numero di *pixel*, l'algoritmo converge alla soluzione più vicina a F_0, fornendo in ogni caso un'approssimazione ragionevole dell'immagine reale.

2.2.1 ART e SIRT

Il metodo **ART (Algebraic Reconstruction Technique)** applica concretamente il *metodo di Kaczmarz* alla ricostruzione tomografica. In ART, i pesi w_{ij} vengono spesso semplificati con valori binari: assumono valore 1 se il centro della cella cade all'interno del raggio, 0 altrimenti. Questa semplificazione facilita il calcolo durante l'esecuzione, ma introduce delle approssimazioni che possono influire sulla qualità della ricostruzione finale.

L'aggiornamento dei *pixel* in ART viene effettuato immediatamente dopo l'elaborazione di ciascun raggio, distribuendo in parti uguali l'errore misurato sui *pixel* attraversati.

La nuova formula di aggiornamento è scritta come:

$$\Delta f_j^{(i)} = \frac{p_i - q_i}{N_i} \tag{2.3}$$

dove:

- p_i è il *ray-sum* misurato dal raggio i.
- q_i è il *ray-sum* stimato dalla ricostruzione corrente (per lo stesso raggio i).
- N_i è il numero di *voxel* attraversati dal raggio i

Una versione più accurata dell'algoritmo sostituisce N_i con L_i, la lunghezza del percorso del raggio all'interno dell'immagine, migliorando la qualità a scapito della complessità computazionale.

Il principale limite di ART riguarda l'aggiornamento immediato dei *pixel*, che può spesso generare artefatti detti '**a sale e pepe**', dovuti principalmente all'inconsistenza tra le equazioni. Ogni raggio modifica le celle già aggiornate dai raggi precedenti, producendo correzioni amplificate e localmente incoerenti.
Per attenuare questo problema, ART utilizza spesso un **fattore di rilassamento** $\alpha < 1$, che riduce l'ampiezza delle modifiche rendendo gli aggiornamenti meno aggressivi. Una strategia comune consiste nell'adottare un valore iniziale elevato di α, per poi diminuirlo progressivamente con l'avanzare delle iterazioni.

L'algoritmo **SIRT (Simultaneous Iterative Reconstruction Technique)** rappresenta un'evoluzione dell'ART[5]. Esso condivide le stesse formule di correzione del metodo precedente, adottando un approccio differente nell'applicazione delle modifiche.

In SIRT, tutte le correzioni dei *pixel* vengono accumulate senza essere immediatamente applicate all'immagine. L'aggiornamento viene eseguito solo alla fine di ogni iterazione, mediando tutte le correzioni raccolte e garantendo un contributo equilibrato di ciascun raggio alla ricostruzione. Questo approccio riduce significativamente il rumore a sale e pepe, producendo delle immagini più uniformi.

Sebbene SIRT richieda un tempo maggiore per convergere rispetto a ART, la sua maggiore qualità e stabilità lo rende adatto a scenari con dati rumorosi o sottodeterminati.

2.3 Reti neurali convoluzionali

Negli ultimi anni le **reti convoluzionali (CNN)** hanno rivoluzionato il campo della visione artificiale, superando lo stato dell'arte nel riconoscimento e nella classificazione delle immagini[6]. Nonostante le CNN esistano da tempo, solo da pochi anni è stato possibile utilizzarle su larga scala, grazie principalmente a due fattori: la disponibilità di grandi *dataset* (come per esempio *ImageNet*) e l'incremento della capacità di calcolo, che ha permesso l'addestramento di reti profonde con milioni di parametri.

Le CNN sono particolarmente utili nei problemi di **segmentazione pixel-wise**, dove è necessario distinguere elementi specifici all'interno di un'immagine. Come specificato nella sezione 2.2, i problemi di ricostruzione tomografica vengono spesso affrontati con algoritmi iterativi, i quali possono soffrire di artefatti locali, dovuti allo scarso numero di proiezioni disponibili.

In questo contesto, le CNN possono essere sfruttate per:

- accelerare la convergenza degli algoritmi iterativi, fornendo una stima iniziale più vicina alla soluzione reale;
- ridurre artefatti e rumore nelle immagini ricostruite (come in questa tesi);
- integrare informazioni di contesto globale e locale.

All'interno della rete, le immagini vengono considerate come **matrici bidimensionali (o tridimensionali)**, in cui ogni elemento rappresenta l'intensità di un *pixel* (o *voxel*). Esse vengono elaborate tramite **filtri convolutivi**, ossia piccole matrici (**kernel**) che scorrono sull'immagine originale. Ogni filtro è addestrato per riconoscere determinati *pattern*, come bordi, angoli o specifiche *texture*[7].

In pratica, il filtro analizza un piccolo blocco di *pixel*, pesando i valori raccolti in base ai coefficienti del filtro applicato e calcolandone la somma. Questa operazione genera una **feature map**, che evidenzia dove uno specifico *pattern* è presente all'interno dell'immagine (Fig. 2.2).

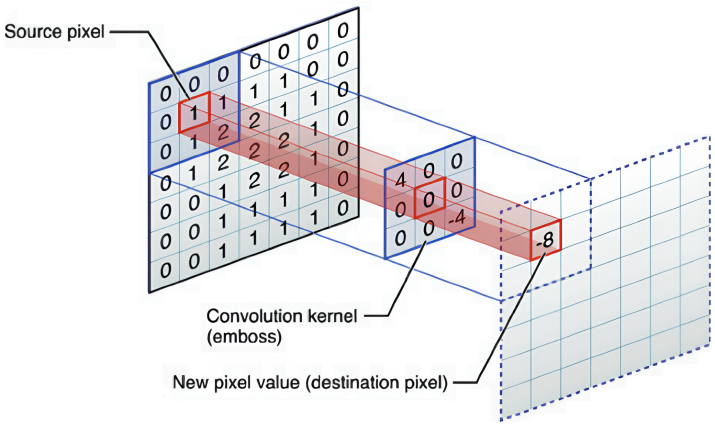


Figura 2.2: Esempio del processo di calcolo di un singolo elemento della *feature map*

Le reti convoluzionali sono organizzate in una serie di strati (**layer**), che elaborano l'input in modo progressivo. Nei livelli iniziali vengono applicati filtri molto semplici, capaci di mettere in evidenza elementi di base, come contorni e linee, mentre gli strati più profondi rielaborano e combinano le informazioni raccolte nei livelli precedenti, costruendo delle rappresentazioni più astratte e ricche di dettagli.

Questa gerarchia ritorna anche nel contesto della ricostruzione tomografica: gli strati iniziali mettono in evidenza bordi e linee fondamentali, mentre quelli più profondi imparano a riconoscere pattern complessi e a distinguere i dettagli, effettivamente legati alla struttura reale, dagli artefatti generati dalla scarsità di dati.

2.3.1 U-Net

La **U-Net**[8] è una rete convoluzionale di tipo **encoder-decoder**, originariamente progettata per la segmentazione di immagini mediche. Il concetto di *encoder-decoder* prevede due fasi principali: nella prima fase si riduce progressivamente la dimensione spaziale dell'immagine attraverso operazioni di **downsampling**, con l'obiettivo di estrarre *features* sempre più astratte dei *pattern* individuati; nella seconda, invece, si espandono queste *features* tramite operazioni di **upsampling**, ricostruendo così un output della stessa dimensione dell'input.

Nella struttura originale dell'U-Net, l'output finale era leggermente più piccolo dell'input, poiché le convoluzioni utilizzate erano di tipo *valid* — cioè senza *padding* e applicate solo alle porzioni dell'immagine completamente coperte dal filtro — provocando, ad ogni strato convolutivo, la perdita di alcuni *pixel* ai bordi (Vedi Fig. 2.3).

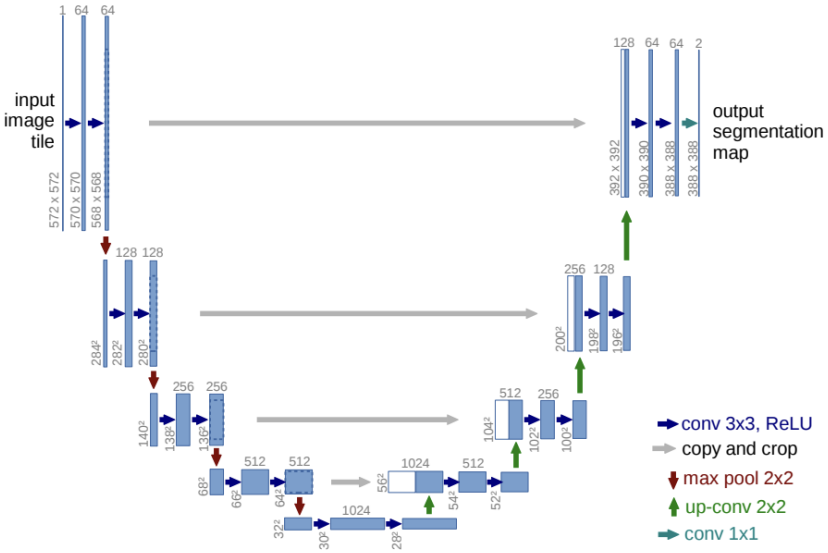


Figura 2.3: Struttura originale dell'architettura U-Net

Durante le operazioni di *downsampling*, parte delle informazioni spaziali può venire persa, rendendo più complicata la ricostruzione precisa dei bordi e delle linee. Per ovviare a questo problema, la U-Net introduce connessioni dirette tra i livelli di *encoder* e *decoder*, che prendono il nome di **skip connections**. Questi collegamenti trasferiscono informazioni ad alta risoluzione direttamente ai livelli di *upsampling*, aumentando la precisione della ricostruzione.

Tuttavia, la U-Net presenta delle criticità importanti quando l'architettura è molto profonda: la problematica principale è la **scomparsa del gradiente (vanishing gradient problem)**.
Durante l'addestramento, l'aggiornamento dei pesi — che avviene durante la fase di *backpropagation* — è basato sul gradiente della funzione di perdita. Nelle architetture molto profonde, il gradiente tende a ridursi man mano che si propaga all'indietro verso i primi *layer*, arrivando quasi ad annullarsi. Questo comportamento rende l'addestramento molto lento e instabile e, di conseguenza, la rete rischia di non convergere più ad una buona soluzione.

2.3.2 Residual U-Net

La **Residual U-Net** è un'evoluzione della U-Net tradizionale e affronta il problema della scomparsa del gradiente integrando **blocchi residuali (residual block)** all'interno dell'architettura *encoder-decoder*[9].

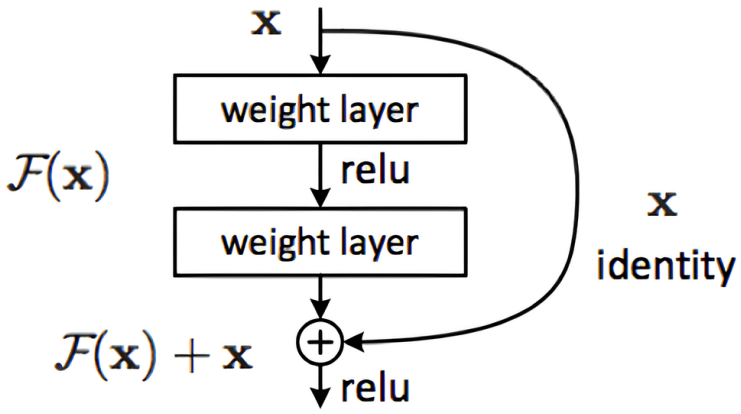


Figura 2.4: Struttura di un blocco residuale

Nell'U-Net classica, ogni blocco prende l'input, lo elabora tramite una serie di convoluzioni e produce una *feature map* che sostituisce l'input di partenza.
Nella Residual U-Net, invece, ogni blocco convolutivo viene sostituito da un **blocco residuale**, che non si limita a trasformare i dati in ingresso mediante convoluzioni, ma li inoltra anche in parallelo tramite un collegamento diretto detto **shortcut connection** (Fig. 2.4).
L'output finale è dato, quindi, dalla somma tra il risultato delle convoluzioni e l'input di partenza.

Grazie a questa architettura, ogni blocco non deve più imparare una trasformazione completa, ma solo una 'correzione' (**residuo**) da applicare all'input. Questo approccio favorisce una propagazione del gradiente più stabile in quanto non dipende unicamente dalle convoluzioni, ma riceve sempre un contributo diretto dall'input originale. Queste caratteristiche permettono alla Residual U-Net di essere più profonda e precisa, senza risentire del problema della scomparsa del gradiente.

Per tale motivo, questa architettura risulta particolarmente efficace per le ricostruzioni tomografiche, migliorando la capacità di distinguere strutture reali dagli artefatti.

Capitolo 3

Caso di Studio

In questa tesi, l'oggetto in esame non è una semplice immagine bidimensionale, ma un volume tridimensionale con dimensioni paragonabili a quelle di un bagaglio. Gli algoritmi utilizzati nella simulazione operano direttamente sui *voxel*, ossia l'equivalente tridimensionale dei *pixel*. I principi teorici illustrati nei capitoli precedenti restano validi: ciò che cambia è il passaggio dalla rappresentazione bidimensionale a quella tridimensionale.

Nel presente capitolo viene inizialmente descritto il processo di generazione del dataset utilizzato per le simulazioni. Successivamente, vengono analizzate in dettaglio le geometrie già implementate nella libreria ASTRA Toolbox e viene introdotto il concetto di geometria *custom*, poi ripreso nella simulazione del processo di screening aeroportuale. Si mostra, quindi, il procedimento per realizzare una geometria di questo tipo, capace di riprodurre le condizioni tipiche dello screening, e si descrive l'implementazione dell'algoritmo di ricostruzione SIRT, utilizzato per ricostruire i volumi originali. Viene presentato, infine, il modello di Residual U-Net utilizzato per la rimozione degli artefatti dalle ricostruzioni ottenute, fornendo una panoramica delle sue caratteristiche e del processo di *training* adottato.

L'obiettivo del capitolo è illustrare, passo dopo passo, il processo necessario per simulare uno screening aeroportuale, combinando algoritmi di ricostruzione algebrici con tecniche di *deep learning*.

3.1 Generazione del dataset

Per le simulazioni eseguite è stato creato un dataset sintetico composto da volumi tridimensionali contenenti ellissoidi e punti di diverse dimensioni e opacità. Ogni volume è inizialmente definito come una matrice di zeri di dimensione $(X,\ Y,\ Z)$, a cui vengono aggiunte figure geometriche secondo regole casuali, ottenendo così un insieme vario e realistico.

In particolare, per ciascun volume, vengono aggiunti:

- **da 5 a 9 ellissi** di opacità casuale, con centro estratto in modo uniforme all'interno del volume. Gli assi principali sono scelti in un intervallo proporzionale alle dimensioni complessive, e l'orientamento è dato da angoli casuali tra 0° e 90° nelle tre direzioni.
- **da 5 a 9 circonferenze** tridimensionali, di raggio compreso tra 3 e 8 *voxel* e con opacità fissa pari a 0.9, per simulare piccoli oggetti densi.

Alcune delle ellissi già inserite vengono inoltre utilizzate come centri per aggiungere ulteriori ellissi concentriche, di dimensioni leggermente maggiori, al fine di creare strutture più complesse.

La **fusione tra due o più figure** avviene tramite una funzione di unificazione, che conserva l'oggetto più opaco nelle regioni di sovrapposizione: quando due o più oggetti si intersecano, prevale quello che rappresenta un materiale più denso. Al termine della generazione, i volumi risultanti vengono raccolti in un tensore di forma $(N,\ X,\ Y,\ Z)$, dove N è il numero di volumi.

Il risultato è un dataset artificiale, utile per simulare la presenza di oggetti di varia forma e densità all'interno di volumi tridimensionali, in analogia a ciò che avviene nello screening dei bagagli (Fig. 3.1).

I volumi generati sono stati poi utilizzati come oggetti virtuali su cui simulare la proiezione dei raggi, fornendo l'input al processo di ricostruzione in ASTRA, secondo il procedimento indicato nella Sezione 3.3. Successivamente, le coppie *(volume originale, volume ricostruito con ASTRA)* sono state utilizzate per allenare un modello di Residual U-Net mirato a rimuovere gli artefatti generati, il cui funzionamento è descritto nella Sezione 3.4.

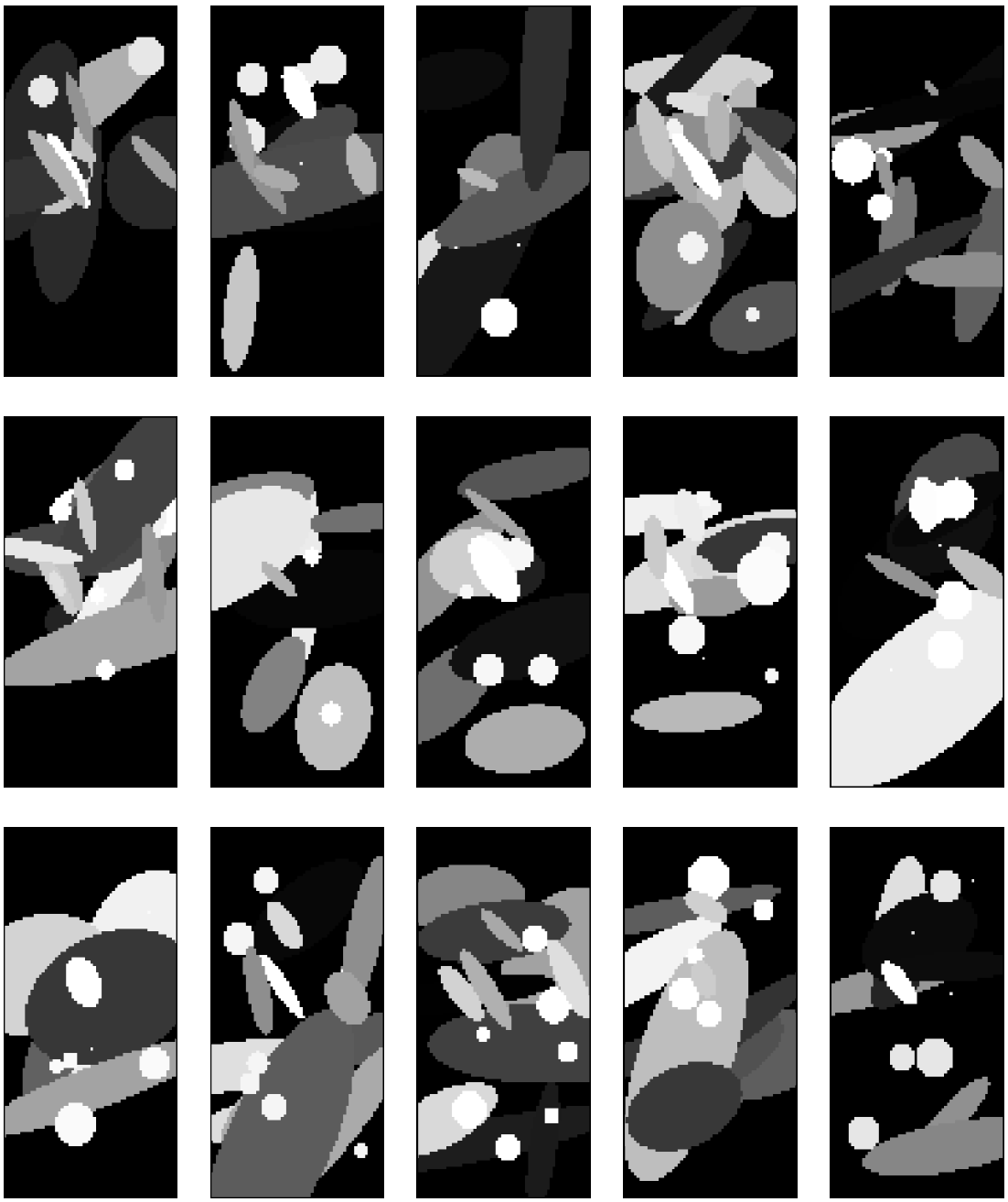


Figura 3.1: Esempi di volumi generati, visualizzati nella loro sezione assiale. Ogni volume è ottenuto combinando forme geometriche con posizioni, dimensioni e opacità variabili, al fine di riprodurre strutture volumetriche complesse e diversificate.

3.2 Geometrie in ASTRA

Per eseguire correttamente una simulazione tomografica tridimensionale, è necessario definire con precisione la geometria di acquisizione.
ASTRA offre diverse geometrie predefinite e la possibilità di crearne di personalizzate. Questa flessibilità è fondamentale per adattare la simulazione al caso specifico che si desidera analizzare.

In questa sezione si presentano le principali geometrie disponibili all'interno della libreria e si introduce il concetto di geometria *custom*.

3.2.1 Geometria parallel3d

La geometria **parallel3d** rappresenta l'estensione tridimensionale della geometria parallela bidimensionale. In questa configurazione, tutti i raggi sono paralleli tra loro, ovvero condividono la stessa direzione di propagazione. Le proiezioni vengono acquisite lungo direzioni fisse, che solitamente ruotano attorno all'oggetto da ricostruire; ciascuna proiezione rappresenta una sezione trasversale del volume.

Questa geometria costituisce un **modello ideale**, difficilmente realizzabile, ed è impiegata principalmente in studi teorici e simulazioni numeriche, consentendo di analizzare il comportamento degli algoritmi senza l'influenza di artefatti sperimentali.

3.2.2 Geometria cone-beam

La geometria **cone-beam** rappresenta, invece, l'estensione tridimensionale della geometria *fan-beam*. In questo modello, i raggi si propagano da una sorgente puntiforme, formando un fascio conico che colpisce il detector. Anche in questo caso, la sorgente e il detector ruotano attorno all'oggetto per acquisire proiezioni da diverse angolazioni.

La traiettoria di ciascun raggio può essere descritta dalla seguente relazione:

$$r(u, v) = s + \lambda \cdot d(u, v), \tag{3.1}$$

dove s è il punto di partenza del raggio (la posizione della sorgente), $d(u, v)$ è un vettore che indica la direzione di propagazione e λ è la distanza percorsa lungo tale direzione.

Questo modello, oltre a essere più realistico rispetto alla geometria *parallel3d*, garantisce una maggiore efficienza di acquisizione, poiché ogni singolo angolo di proiezione permette di campionare simultaneamente una porzione tridimensionale più ampia del volume (Vedi fig. 3.2).

Tale vantaggio si accompagna, però, a una maggiore complessità computazionale, dovuta alla gestione della divergenza e alla necessità di effettuare correzioni geometriche.

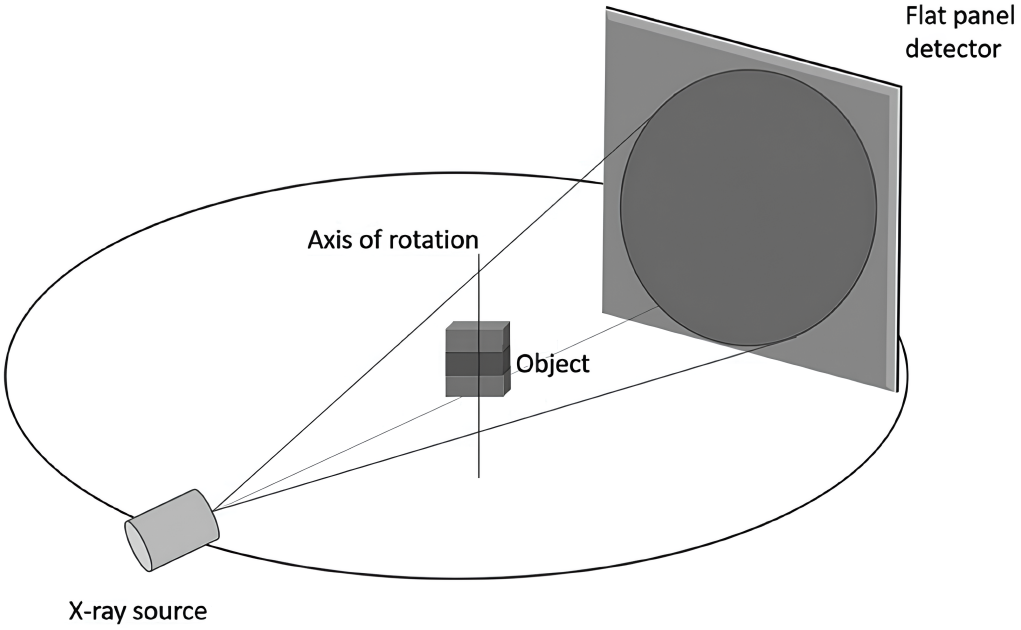


Figura 3.2: Immagine che descrive il funzionamento di una geometria *cone-beam*. La sorgente puntiforme emette un fascio conico di raggi che attraversa l'oggetto, ruotando insieme al detector attorno a esso per acquisire proiezioni tridimensionali da diverse angolazioni.

3.2.3 Geometrie custom

ASTRA permette anche di definire geometrie completamente personalizzate tramite i tipi **parallel3d_vec** e **cone_vec**. Questi tipi mantengono le caratteristiche fondamentali delle loro controparti non custom - *parallel3d* e *cone* - e, allo stesso tempo, consentono di specificare per ciascuna proiezione: la posizione della sorgente, il centro del detector e l'orientamento di quest'ultimo.

Ogni passo di acquisizione è quindi definito da un vettore di 12 valori che specifica in ordine:

- **Valori 1–3:** la **sorgente** (x, y, z)
- **Valori 4–6:** il **centro del detector** (x, y, z)
- **Valori 7–9: l'asse orizzontale** del detector (x, y, z)
- **Valori 10–12: l'asse verticale** del detector (x, y, z)

Questa flessibilità consente di simulare configurazioni sperimentali complesse, inclusi movimenti non rotazionali, che non possono essere riprodotti fedelmente con le geometrie predefinite.

3.3 Simulazione dello screening

In questa sezione viene descritto il processo di simulazione realizzato con ASTRA Toolbox.

Per modellare in maniera realistica l'acquisizione dei bagagli, è stata inizialmente adottata la geometria *cone*, pensata originariamente per sistemi di tomografia rotazionale. Tuttavia, nel contesto dello screening aeroportuale la situazione è diversa: il bagaglio non resta fermo mentre sorgente e detector ruotano, ma si muove su un nastro trasportatore attraverso un sistema di acquisizione stazionario. Per riprodurre questo comportamento, è necessario implementare una tomografia di tipo traslazionale (Sezione 1.5) tramite una geometria custom di tipo **cone_vec**.

Inizialmente, si definiscono i parametri di acquisizione, che comprendono:

- **durata dell'esposizione** (in secondi), ossia il tempo impiegato dall'oggetto per percorrere il nastro;
- **frequenza di campionamento** (in Hz), ovvero il numero di proiezioni eseguite per secondo;
- **dimensioni del bagaglio** simulato;
- **caratteristiche del sistema di rivelazione** (numero di sorgenti, numero e dimensione dei detector, distanze sorgente-oggetto e oggetto-detector).

A partire da questi parametri si calcolano il **numero totale di proiezioni** (N) e la **lunghezza del nastro simulato** (s), che in questo contesto corrisponde allo spazio percorso dalle sorgenti e dai detector.

Le formule utilizzate sono le seguenti:

$$N = f \cdot t \tag{3.2}$$

$$s = 2\left(\frac{L_d}{2} + \frac{L_o}{2}\right) + \Delta \tag{3.3}$$

dove:

- f è la frequenza di campionamento (Hz);

- t è la durata dell'esposizione (s);
- L_d è la lunghezza del detector lungo l'asse di traslazione;
- L_o è la lunghezza dell'oggetto lungo l'asse di traslazione;
- Δ è un'*offset* aggiuntivo, utile per evitare che l'oggetto sia in parte all'interno del campo visivo del detector già nelle prime proiezioni (migliorando così la qualità dei dati raccolti).

Dopo aver impostato i parametri iniziali, si definiscono i vettori della geometria *custom*. Per la simulazione, sorgente e detector vengono traslati lungo un asse di traslazione (x o y), mantenendo il volume fisso al centro. In altre parole, ad ogni passo, la sorgente e il detector avanzano di un certo numero di unità, replicando il movimento del bagaglio sul nastro. Questa soluzione permette di ottenere la stessa acquisizione che avverrebbe nella realtà. Va tuttavia fatta una precisazione: una singola coppia sorgente-detector fornisce informazioni limitate e, dunque, non sufficienti per ricostruire il volume in modo accurato (Fig. 3.3).

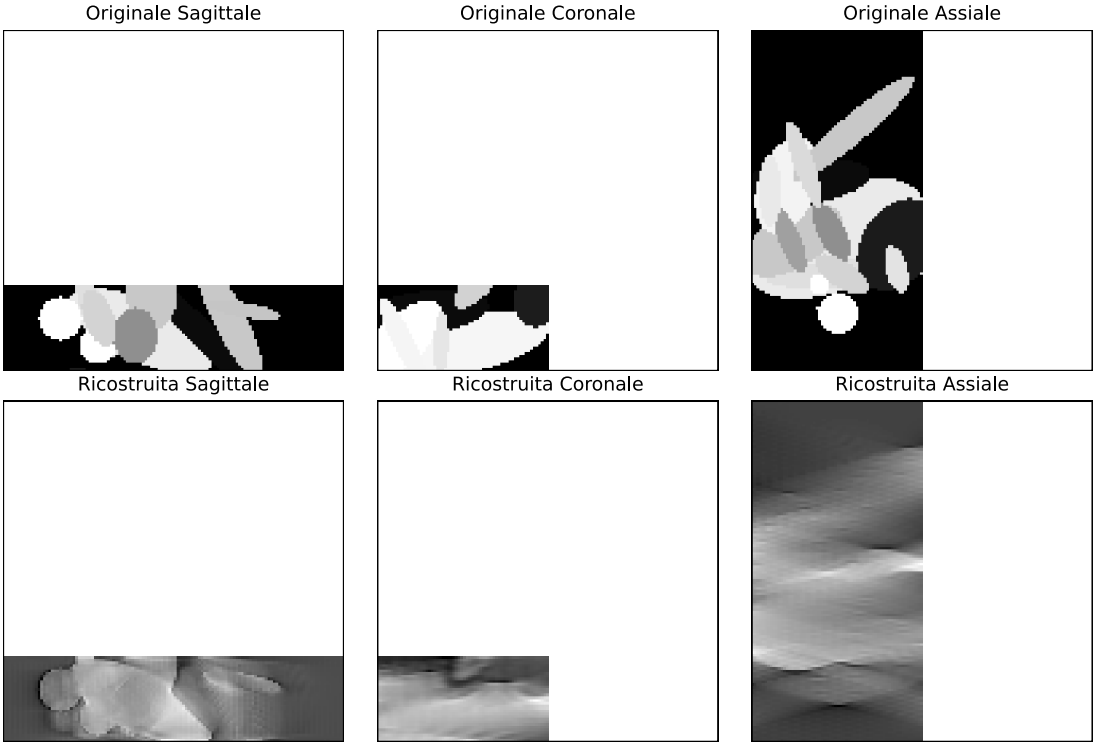


Figura 3.3: Simulazione della ricostruzione di un volume utilizzando una singola sorgente laterale. È evidente come un unico punto non sia sufficiente per ricostruire correttamente le altre due viste, poiché queste non vengono attraversate direttamente dai raggi.

Per questo motivo si utilizzano due coppie:

- una collocata in alto, che cattura informazioni sulla **sezione assiale** (dall'alto al basso),
- una collocata lateralmente, che cattura informazioni sulla **sezione sagittale** (da sinistra a destra).

Combinando i dati acquisiti da queste due coppie è possibile ricostruire anche la terza sezione, la **coronale**, ottenendo un volume tridimensionale molto vicino a quello originale.

Il set completo di vettori geometrici avrà quindi dimensione $(N \times 2, 12)$.
Per ciascun passo i, vengono definite due configurazioni di valori:

- **Sorgente laterale**
  - sorgente: $(p_i, -SO_1, 0)$
  - centro detector: $(p_i, OD_1, 0)$
  - asse orizzontale: $(1, 0, 0)$
  - asse verticale: $(0, 0, 1)$
- **Sorgente dall'alto:**
  - sorgente: $(p_i, 0, SO_2)$
  - centro detector: $(p_i, 0, -OD_2)$
  - asse orizzontale: $(1, 0, 0)$
  - asse verticale: $(0, 1, 0)$

dove:

- SO_1 e SO_2 rappresentano le distanze *sorgente-oggetto* delle due viste;
- OD_1 e OD_2 rappresentano le distanze *oggetto-detector* delle due viste;
- p_i indica la posizione della sorgente e del detector al passo i lungo l'asse di traslazione.

In questo modo, per ogni posizione p_i sul nastro traslazionale, vengono generate due proiezioni indipendenti: una laterale e una dall'alto, che insieme forniscono la copertura necessaria (Fig. 3.4).

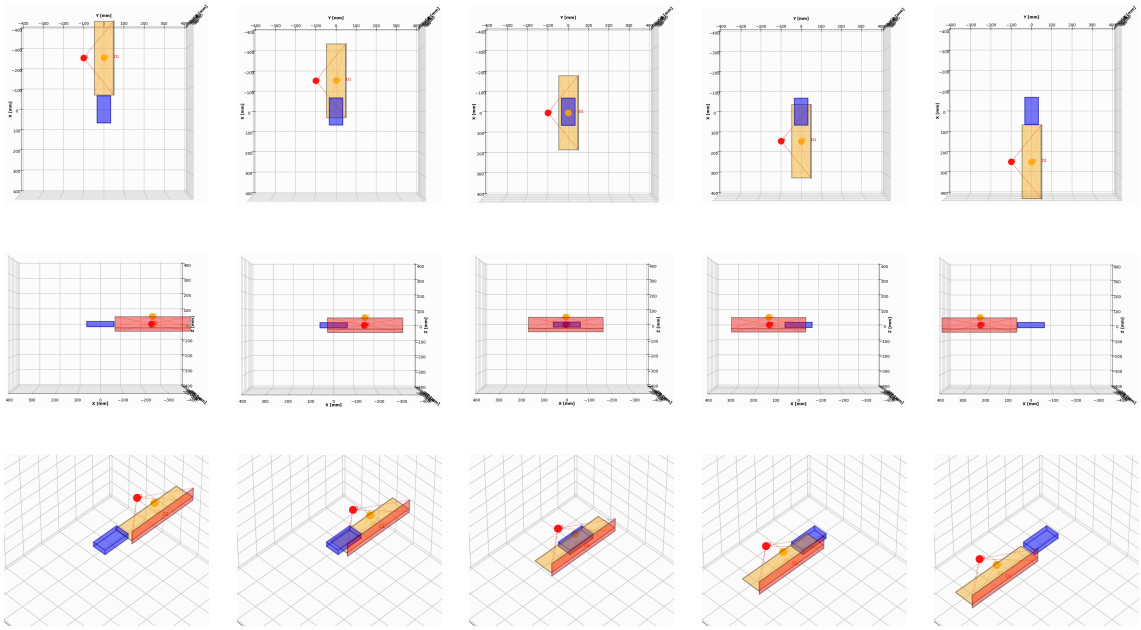


Figura 3.4: Frame che mostrano la traslazione delle coppie sorgente-detector lungo il piano di acquisizione nei diversi passi di proiezione. La sorgente laterale è rappresentata in rosso, mentre quella dall'alto è evidenziata in giallo. La figura in blu rappresenta il bagaglio.

Le posizioni di campionamento p_i sono distribuite uniformemente nell'intervallo $[-s/2,\, s/2]$ secondo la relazione:

$$p_i = -\frac{s}{2} + i \cdot \frac{s}{N}, \qquad i = 0, 1, \ldots, N-1 \tag{3.4}$$

dove s indica la lunghezza del nastro simulato e N il numero totale di proiezioni.

Definita l'intera geometria di proiezione, il volume viene sottoposto all'algoritmo **FP3D_CUDA**, che simula una proiezione in avanti (*forward projection*). In altre parole, l'algoritmo calcola come i raggi X attraverserebbero il volume lungo le direzioni definite dai vettori geometrici, generando così il sinogramma corrispondente alle proiezioni simulate.

A partire dal sinogramma, si procede con la fase di ricostruzione utilizzando l'algoritmo **SIRT3D_CUDA**, versione tridimensionale del metodo SIRT. La scelta è ricaduta su questo algoritmo per due motivi principali: da un lato, è già implementato e ottimizzato per ASTRA, garantendo un'esecuzione efficiente e una simulazione fedele; dall'altro, è tuttora considerato uno degli algoritmi di ricostruzione più affidabili e performanti in circolazione.

Una volta ricostruito il volume, è possibile valutare la qualità della ricostruzione confrontando il volume ricostruito con quello originale, come verrà illustrato nel capitolo 4.

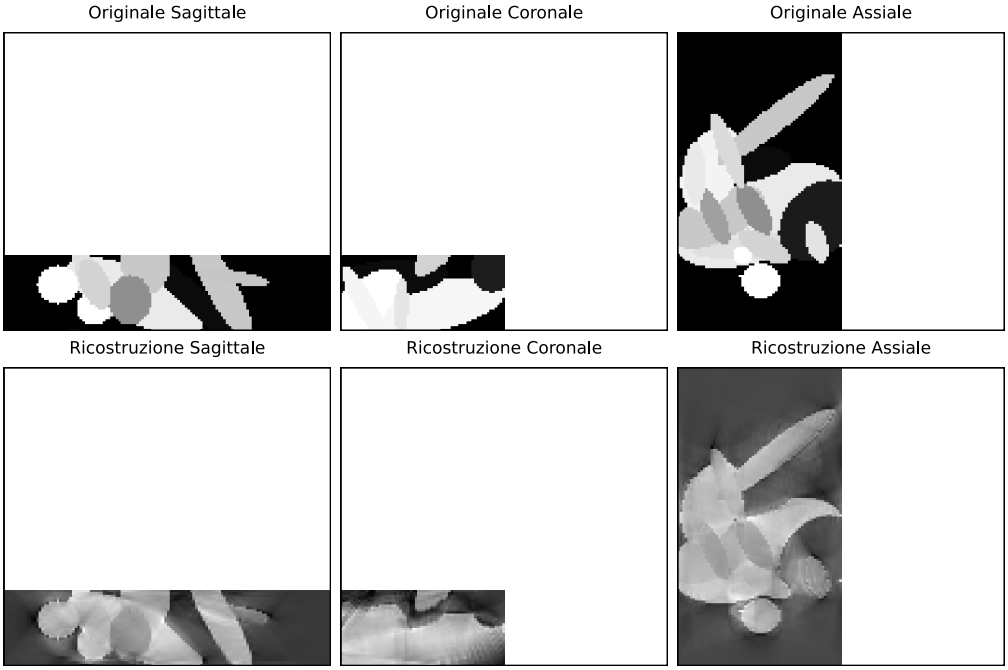


Figura 3.5: Sezioni rappresentative di un volume ricostruito mediante il procedimento descritto in precedenza.

Osservando i risultati della fig. 3.5, si nota come la ricostruzione riproduca con buona precisione i contorni degli oggetti contenuti all'interno del bagaglio, pur presentando numerosi artefatti che ne compromettono la qualità complessiva.

3.4 Rimozione degli artefatti

Come osservato nella Sezione 3.3, la ricostruzione ottenuta con SIRT3D_CUDA presenta artefatti significativi, dovuti principalmente al numero limitato di proiezioni disponibili e alle caratteristiche della geometria traslazionale, che non consente una copertura uniforme del volume.

Per migliorare la qualità della ricostruzione, è stato adottato un approccio basato sul deep learning, con l'obiettivo di rimuovere gli artefatti mantenendo al contempo intatte le informazioni strutturali rilevanti (vedi Fig. 3.6)

Il modello sviluppato è una **Residual U-Net**, progettata specificamente per l'elaborazione di dati volumetrici. Essa è caratterizzata, come la U-Net classica, da una fase di **encoding** e una di **decoding**.

L'**encoder** è costituito da tre livelli di *downsampling*. In ciascun livello, la dimensione del volume in ingresso viene ridotta tramite operazioni di *pooling* e convoluzioni, producendo rappresentazioni sempre più compatte dell'oggetto. Ad ogni riduzione, inoltre, la rete incrementa il numero di *feature map*, aumentando così la capacità del modello di descrivere aspetti globali della struttura.

Il **decoder** presenta una struttura simmetrica con tre livelli di *upsampling*. In questa fase, la risoluzione del volume viene progressivamente ripristinata tramite operazioni di interpolazione (o *traspose convolutions*). A ogni livello, il numero di *feature map* si riduce e le nuove rappresentazioni vengono concatenate con quelle corrispondenti dell'encoder attraverso l'utilizzo di *skip connections*. In questo modo si combinano le informazioni ottenute e si recuperano i dettagli locali persi durante la fase di *downsampling*.

Ogni layer nell'encoder e nel decoder utilizza un blocco residuale. Questo blocco è composto da due convoluzioni tridimensionali $3 \times 3 \times 3$, seguite dalla funzione di attivazione **ReLU**. Inoltre, l'input originale viene sottoposto a una convoluzione $1 \times 1 \times 1$ per renderlo compatibile con l'output prodotto. Grazie ai blocchi residuali, la rete può concentrarsi sull'apprendere le differenze tra l'input e l'output desiderato, semplificando il processo di addestramento.

Al termine del decoder, si applica un'ulteriore convoluzione $1 \times 1 \times 1$, che riduce i canali delle *feature map* a uno, seguita da un'attivazione **lineare**. In questo modo, tutte le informazioni ottenute vengono condensate in un'unica **rappresentazione finale**.

Le **attivazioni** sono funzioni matematiche, applicate all'uscita di ogni layer, che introducono non linearità e permettono alla rete di rappresentare relazioni complesse tra i dati. Nelle fasi di encoding e decoding viene impiegata la ReLU per estrarre e trasformare le *features*, mentre nella fase finale, dove l'intento è quello di ricostruire il volume con valori di intensità continui per ogni *voxel*, si impiega un'attivazione lineare, che mantiene inalterati i valori prodotti dalla convoluzione.

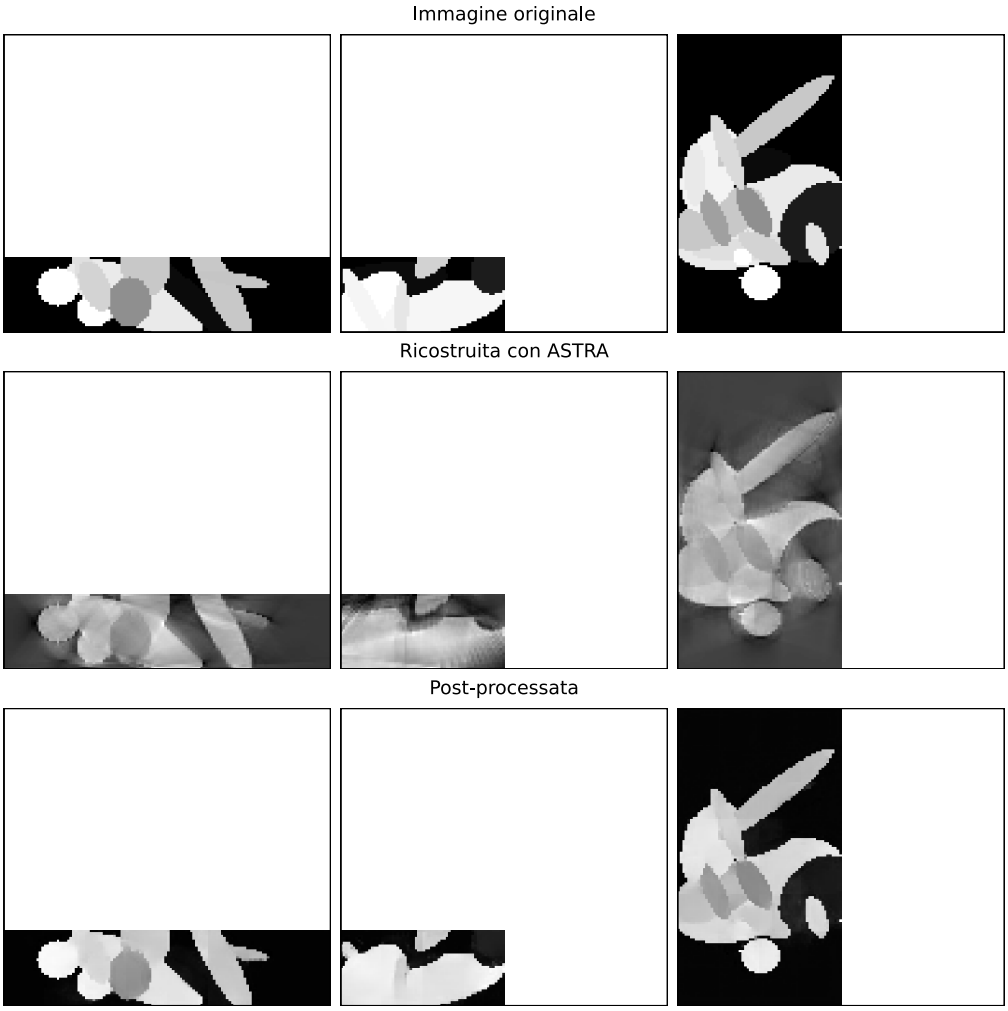


Figura 3.6: Esempio di rimozione degli artefatti mediante il modello Residual U-Net. Le tre immagini mostrano le sezioni sagittale, coronale e assiale del volume ricostruito, evidenziando una migliore qualità dei dettagli rispetto alla ricostruzione di partenza con ASTRA.

3.4.1 Divisione in patches

Per evitare che il modello si abitui a ricostruire sempre volumi della stessa dimensione, l'addestramento viene eseguito su piccole **patch tridimensionali** estratte dai volumi. In questo modo, la rete impara a lavorare su regioni locali e può essere applicata a volumi di dimensioni diverse.

Dato un volume di dimensioni ($X,\ Y,\ Z$), esso viene diviso in blocchi di dimensione ($p_x,\ p_y,\ p_z$), utilizzando uno **stride** ($s_x,\ s_y,\ s_z$). La dimensione dello *stride* definisce il passo di spostamento lungo ciascuna direzione: in altre parole, indica di quanti *voxel* ci si sposta lungo ogni asse prima di estrarre il blocco successivo. Se le dimensioni originali non sono multipli esatti della dimensione della patch o dello stride, il volume viene esteso con **padding simmetrico**, cioè aggiungendo *voxel* nulli ai bordi in modo da raggiungere le dimensioni necessarie, preservando la posizione del volume centrale. Rimossi gli artefatti in ciascuna *patch*, il modello procede alla ricostruzione del volume ricombinando i blocchi. Nelle aree in cui più *patch* si sovrappongono, i valori dei *voxel* vengono mediati, al fine di garantire transizioni fluide e minimizzare gli artefatti derivanti dalla suddivisione.

Questa procedura offre due vantaggi principali:

- riduce notevolmente il consumo di memoria sulla GPU, rendendo possibile l'addestramento di volumi di grandi dimensioni;
- introduce una maggiore variabilità nei dati, poiché ogni *patch* rappresenta una porzione locale diversa del volume.

3.4.2 Data augmentation

La tecnica di **data augmentation** è comunemente impiegata nell'addestramento delle reti neurali con lo scopo di incrementare la variabilità del dataset e ridurre il rischio di overfitting. Consiste nell'applicare trasformazioni casuali ai dati in input (*ground truth*), in modo che il modello impari a riconoscere le strutture indipendentemente dal loro orientamento o dalla loro disposizione spaziale. In questo modo, la rete viene esposta a scenari sempre diversi e acquisisce una maggiore capacità di generalizzazione sui dati[10].

Nel caso in esame è stata adottata una strategia di **data augmentation on the fly**, nella quale i campioni non vengono pre-generati e salvati, ma trasformati dinamicamente ad ogni epoca. In particolare, le *patch* estratte dai volumi subiscono operazioni di **flip** lungo gli assi principali con una probabilità del 50% e **rotazioni** casuali di 90° sui piani bidimensionali ($xy,\ xz,\ yz$).

Queste trasformazioni non modificano il contenuto semantico dei dati, ossia le strutture o le caratteristiche di interesse che il modello deve apprendere, ma ne alterano unicamente la rappresentazione geometrica. In questo modo, la rete è costretta a focalizzarsi sulle proprietà intrinseche delle strutture piuttosto che sulla loro posizione o orientamento.

3.4.3 Funzione di perdita

La **funzione di perdita** è uno strumento fondamentale nel training delle reti neurali, poiché quantifica la differenza tra le predizioni del modello e i valori reali. Essa fornisce un'indicazione quantitativa dell'errore e guida l'aggiornamento dei pesi durante la **backpropagation**.

Matematicamente, la funzione di perdita associa a ciascuna coppia $(\hat{y}, y)$ — predizione e valore reale — un numero scalare che rappresenta l'errore. L'obiettivo del training è minimizzare questa funzione sull'intero **training set**, affinché il modello produca predizioni sempre più accurate.

La scelta della funzione di perdita dipende dal tipo di problema ed è cruciale per garantire una buona capacità di generalizzazione del modello.

3.5 Processo di training

Il **processo di training** può essere descritto come segue:

1. Il dataset viene suddiviso in due sottoinsiemi principali: il **training set**, utilizzato per aggiornare i pesi della rete durante l'addestramento, e il **validation set**, impiegato per monitorare le prestazioni del modello su dati non visti e per prevenire l'overfitting.
2. La fase di training viene suddivisa in **epoche**, ciascuna delle quali corrisponde a un passaggio completo sul *training set*.
3. Per ogni epoca, i dati vengono suddivisi casualmente in sottoinsiemi più piccoli chiamati **batch**. Ogni *patch* all'interno di un *batch* può essere sottoposto a trasformazioni casuali (*data augmentation*).
4. Per ogni *batch*, il modello calcola le predizioni, confronta i risultati con i valori reali (*ground truth*) mediante una funzione di perdita e aggiorna i pesi tramite *backpropagation*.

In sintesi:

- **L'elaborazione a batch** riduce l'uso di memoria e aggiorna i pesi della rete più frequentemente rispetto ad un'elaborazione dell'intero dataset.
- La tecnica di **data augmentation** assicura che il modello non veda mai due volte lo stesso insieme di dati: durante il training, infatti, le *patches* possono subire trasformazioni, apparendo in configurazioni spaziali diverse e migliorando la capacità di generalizzazione del modello.

- La **valutazione sul validation set**, eseguita alla fine di ogni epoca, permette di monitorare le prestazioni della rete e di adottare strategie come **early stopping** (sezione 4.3.1), migliorando ulteriormente la robustezza del modello e prevenendo l'overfitting.

Capitolo 4

Risultati

In questo capitolo vengono presentati i risultati ottenuti dalla simulazione dello screening dei bagagli. Si analizzano, innanzitutto, le dimensioni e i parametri adottati per la proiezione dei raggi e per la successiva ricostruzione del volume originale mediante l'algoritmo SIRT3D_CUDA. Successivamente, la qualità delle ricostruzioni viene valutata mediante specifiche metriche di accuratezza, tenendo conto delle limitazioni discusse nella sezione 3.4.
Si descrive quindi la scelta degli iperparametri adottati per l'addestramento della Residual U-Net, utilizzata per la rimozione degli artefatti. Infine, si analizzano i volumi post-processati dal modello attraverso un duplice approccio: da un lato, un confronto quantitativo basato sulle metriche di accuratezza; dall'altro, un confronto qualitativo tramite l'osservazione diretta delle sezioni volumetriche ricostruite.

L'obiettivo finale è verificare che la combinazione dell'algoritmo SIRT con il modello Residual U-Net consenta di ottenere volumi fedeli alla realtà, riducendo gli artefatti e preservando le informazioni essenziali per il riconoscimento degli oggetti nello scenario applicativo dello screening aeroportuale.

4.1 Metriche per la valutazione

Per valutare la qualità dei volumi ricostruiti, si utilizzano metriche quantitative che stimano l'accuratezza della ricostruzione rispetto al volume originale di riferimento. In questa tesi si adottano quattro metriche di valutazione: **Mean Squared Error (MSE)**, **Mean Absolute Error (MAE o L1)**, **Peak Signal-to-Noise Ratio (PSNR)** e **Structural Similarity Index Measure (SSIM)**.
Di seguito se ne descrivono le principali caratteristiche.

4.1.1 Mean Squared Error

La metrica Mean Squared Error (MSE) è definita come la media dei quadrati delle differenze tra i *voxel* del volume originale e quelli del volume ricostruito.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (v_i - \hat{v}_i)^2 \tag{4.1}$$

dove v_i e $\hat{v}_i$ sono rispettivamente i valori dei *voxel* del volume originale e di quello ricostruito, e N è il numero totale di *voxel*.

Il valore di MSE è sempre ≥ 0 ed è nullo solo in caso di corrispondenza perfetta. Questa metrica è semplice da calcolare e fornisce una misura diretta dell'errore globale, ma non tiene conto della percezione visiva né della coerenza strutturale delle forme, risultando talvolta poco rappresentativa in determinati contesti.

4.1.2 Mean Absolute Error (o L1 loss)

Mean Absolute Error (MAE), noto anche come **L1 loss**, misura la differenza media in valore assoluto tra i *voxel* del volume originale e quelli del volume ricostruito:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |v_i - \hat{v}_i| \tag{4.2}$$

dove v_i e $\hat{v}_i$ rappresentano rispettivamente i valori dei *voxel* del volume originale e di quello ricostruito, e N è il numero totale di *voxel*.

MAE assume valori sempre positivi, raggiungendo lo zero solo in caso di corrispondenza perfetta. Rispetto a MSE, risulta meno sensibile ai valori anomali grazie all'assenza dell'elevamento al quadrato delle differenze, offrendo così una stima più robusta dell'errore medio. Tuttavia, anche questa metrica, resta una misura puramente *voxel-wise*, non tenendo conto della percezione visiva o della coerenza strutturale.

4.1.3 Peak Signal-to-Noise Ratio

La metrica PSNR deriva direttamente dalla MSE e quantifica la qualità della ricostruzione in decibel (dB). È definito come il rapporto logaritmico tra l'intensità massima possibile di un *voxel* e l'errore quadratico medio.

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_V^2}{MSE} \right) \tag{4.3}$$

dove MAX_V rappresenta il valore massimo che può assumere un *voxel*.

Valori superiori a 30 dB indicano generalmente una buona qualità, mentre valori inferiori a 20 dB denotano una ricostruzione scadente. PSNR fornisce un indice sintetico e facilmente comparabile, ma allo stesso tempo condivide le stesse limitazioni della MSE, essendo la sua trasformazione logaritmica.

4.1.4 Structural Similarity Index Measure

La metrica SSIM valuta, invece, la similarità strutturale tra due ricostruzioni volumetriche. L'indice viene calcolato su finestre tridimensionali di *voxel* e tiene conto di tre componenti fondamentali: la **luminosità**, il **contrasto** e la **struttura**.

$$SSIM(V, \hat{V}) = \frac{(2\mu_V\mu_{\hat{V}} + C_1)(2\sigma_{V\hat{V}} + C_2)}{(\mu_V^2 + \mu_{\hat{V}}^2 + C_1)(\sigma_V^2 + \sigma_{\hat{V}}^2 + C_2)} \tag{4.4}$$

dove μ_V e $\mu_{\hat{V}}$ sono le medie locali dei *voxel*, σ_V^2 e $\sigma_{\hat{V}}^2$ le varianze locali, $\sigma_{V\hat{V}}$ la covarianza, e C_1 e C_2 le costanti di stabilizzazione.

L'indice assume tipicamente valori compresi tra 0 e 1, dove 1 indica una corrispondenza strutturale perfetta. A differenza delle metriche precedenti, SSIM è più vicino alla percezione visiva umana, poiché valuta in che misura le strutture locali vengono preservate. Tuttavia, rimane debole nella rilevazione di variazioni non strutturali, che incidono in misura ridotta sulla sua valutazione.

4.2 Ricostruzione con ASTRA

In questa sezione vengono presentati i criteri di scelta dei parametri utilizzati per la simulazione con ASTRA e i relativi risultati.

4.2.1 Parametri di simulazione

La dimensione del bagaglio, utilizzata per la simulazione dello screening, è stata fissata a 128×64×32 *voxel*.

Nei moderni processi di screening, basati sulla tomografia computerizzata, il bagaglio si muove sul nastro trasportatore sotto lo scanner per un intervallo medio di 4-5 secondi. Pertanto, nella simulazione, la durata dell'esposizione è stata fissata a 4 secondi, al fine di riprodurre in modo realistico le condizioni operative.

Per ricostruire un volume tridimensionale in un sistema traslazionale come quello simulato, è necessario acquisire un numero adeguato di proiezioni, generalmente

compreso tra 200 e 400. Di conseguenza, nei test, la frequenza di emissione dei raggi (*shot frequency*) è stata variata nell'intervallo 50-100 Hz, con lo scopo di valutare diverse configurazioni e determinare un equilibrio ottimale tra qualità e tempo di acquisizione.

Le distanze SO_1, OD_1, SO_2 e OD_2 sono state impostate rispettivamente a 100, 50, 48 e 24 *voxel* (Fig. 4.1). Questo assetto geometrico è stato scelto per assicurare che l'intero volume del bagaglio risultasse visibile in almeno una proiezione e che le sezioni meno esposte non fossero completamente oscurate. In questo modo si ottiene una copertura angolare più uniforme, che riduce il rischio di artefatti di ricostruzione dovuti a zone parzialmente occluse e consente di preservare meglio i dettagli delle strutture interne.

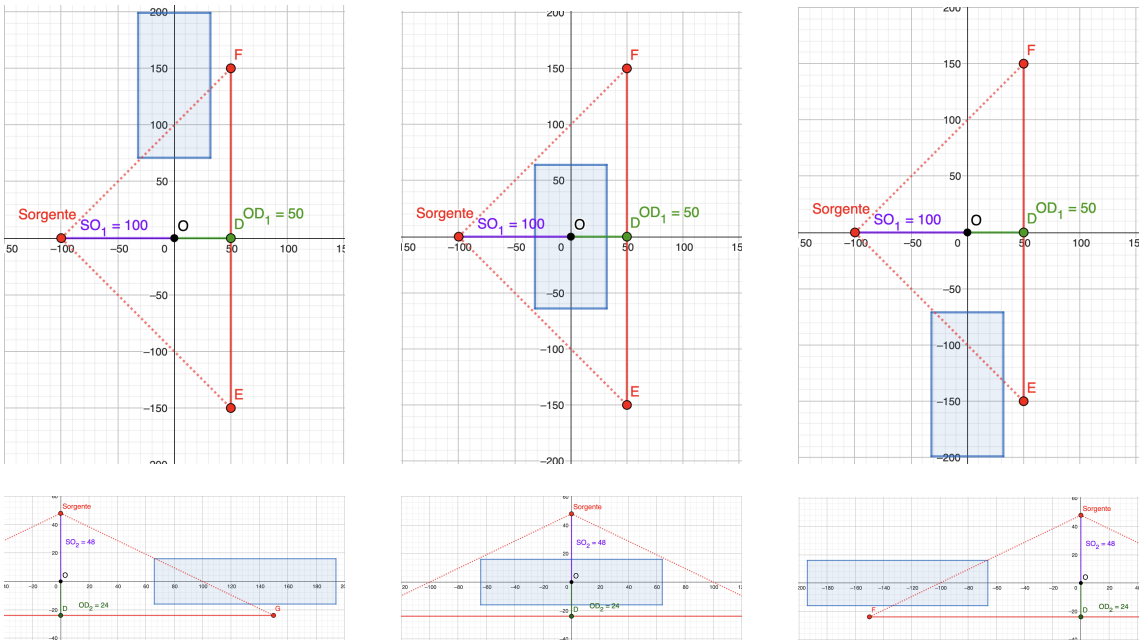


Figura 4.1: Rappresentazione della sorgente laterale (in alto) e di quella dall'alto (in basso), nella configurazione definita in precedenza. L'immagine centrale mostra come l'intero volume risulti completamente visibile in almeno una proiezione, mentre quelle laterali evidenziano una buona copertura angolare anche per le sezioni non direttamente osservabili.

Il numero di iterazioni ottimale per l'algoritmo di ricostruzione SIRT è stato determinato attraverso prove sperimentali. È stato osservato che, già dopo 50 iterazioni, si ottengono ricostruzioni con contorni ben definiti e sufficientemente accurate per essere sottoposte a post-processing, mentre oltre le 150 iterazioni, i miglioramenti risultano marginali e difficilmente percepibili a occhio nudo (Fig. 4.2).
Pertanto, il valore di riferimento si colloca in questo intervallo.

Tra le metriche considerate, non è stato attribuito particolare peso a MSE, MAE e PSNR, poiché in questa fase preliminare l'attenzione era rivolta principalmente alla corretta ricostruzione delle strutture locali, necessarie per la fase successiva di *post-processing*. Per questo motivo, la metrica SSIM ha assunto un ruolo prioritario, in quanto maggiormente indicata nel validare la preservazione di tali caratteristiche strutturali. Oltre agli indici, anche il riscontro visivo ha rappresentato un criterio fondamentale per valutare la qualità delle ricostruzioni e stabilire se esse fossero accettabili ai fini dell'analisi successiva.

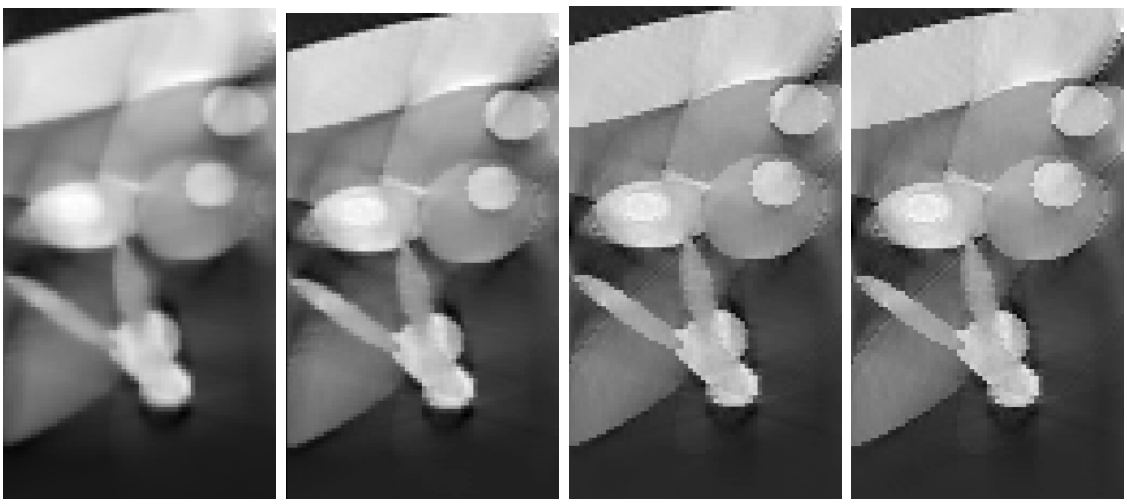


Figura 4.2: Confronto delle ricostruzioni ottenute, a parità di condizioni sperimentali, al variare del numero di iterazioni dell'algoritmo SIRT (da sinistra verso destra: 25, 50, 100 e 150 iterazioni)

4.2.2 Valutazione

Frequenza	Iterazioni	PSNR	SSIM	Tempo per volume (in secondi)
50Hz	25	12.230	0.3597	3.28
50Hz	50	12.236	0.3734	5.82
50Hz	75	12.185	0.3788	8.48
50Hz	100	12.186	0.3828	10.79
50Hz	125	12.212	0.3861	13.35
50Hz	150	12.247	0.3888	15.82
75Hz	25	12.250	0.3603	4.76
75Hz	50	12.251	0.3737	8.42
75Hz	75	12.212	0.3792	12.02
75Hz	100	12.205	0.3831	15.75
75Hz	125	12.219	0.3863	19.53
75Hz	150	12.247	0.3889	23.19
100Hz	25	12.248	0.3605	6.07
100Hz	50	12.244	0.3740	11.08
100Hz	75	12.176	0.3790	15.85
100Hz	100	12.167	0.3828	20.68
100Hz	125	12.196	0.3863	25.66
100Hz	150	12.234	0.3891	30.57

Tabella 4.1: Risultati delle ricostruzioni ottenute con ASTRA Toolbox al variare della frequenza di acquisizione e del numero di iterazioni.

I risultati riportati nella Tabella 4.1 evidenziano come i valori del **PSNR** siano **irregolari** e, in più casi, non coerenti con la qualità effettiva delle ricostruzioni. Tale andamento conferma che, nella fase attuale, questa metrica non rappresenta un indicatore affidabile, poiché fortemente condizionata dalla presenza di artefatti. Al contrario, l'indice **SSIM** si dimostra maggiormente adeguato, in quanto focalizzato sulla preservazione delle strutture del volume e meno sensibile alle sole differenze *voxel* per *voxel*.

Osservando il confronto tra diverse frequenze di acquisizione, a parità di iterazioni, si può notare come i valori di SSIM si mantengano sostanzialmente invariati (si pensi, ad esempio, al caso con 150 iterazioni, dove si ottiene 0.3888 a 50 Hz e 0.3891 a 150 Hz). Ciò suggerisce che una frequenza di **50 Hz**, corrispondente a 200 proiezioni totali, considerando un tempo di permanenza sul nastro pari a 4 secondi, sia già sufficiente a catturare in maniera efficace le informazioni essenziali del volume.

Infine, si osserva un aumento graduale e costante dei valori di SSIM all'aumentare del numero di iterazioni. Tale incremento, tuttavia, risulta quantitativamente

modesto e potrebbe non giustificare l'impiego aggiuntivo di risorse computazionali, né il conseguente aumento di tempo necessario per l'elaborazione di ciascuna ricostruzione.

In conclusione, un buon compromesso tra qualità della ricostruzione e tempi computazionali può essere ottenuto adottando una configurazione a 50 Hz con un numero di iterazioni compreso tra 50 e 100. Questa scelta offre un indice SSIM vicino ai valori massimi osservati, riducendo contemporaneamente il tempo di elaborazione per volume.

Tuttavia, per stabilire con certezza quali parametri siano effettivamente più efficienti, è indispensabile valutare, partendo da queste ricostruzioni, i risultati finali ottenuti dopo la fase di *post-processing* dedicata alla rimozione degli artefatti.

4.3 Rimozione degli artefatti con ResNet

A partire dai dati riportati nella Tabella 4.1, ci concentriamo ora sull'analisi della rimozione degli artefatti, esaminando i risultati ottenuti mediante la Residual U-Net e valutando in che misura una ricostruzione iniziale di qualità superiore, ottenuta con SIRT, possa influenzare il risultato finale.

4.3.1 Iperparametri

Per l'addestramento del modello è stata effettuata una scelta accurata degli **iperparametri**, bilanciando esigenze computazionali, capacità di generalizzazione e qualità delle predizioni.

Il training è stato eseguito utilizzando **patches** di dimensione $32 \times 32 \times 32$. Questa scelta è stata motivata da due esigenze principali: da un lato, la necessità di garantire la compatibilità con le operazioni di **data augmentation** (rotazioni e flip), resa possibile dall'uguaglianza delle tre dimensioni delle patch; dall'altro, la volontà di adattarsi alle dimensioni del volume analizzato: poiché 32 è divisore di 64 e 128, è possibile suddividere il volume (di dimensione $128 \times 64 \times 32$) senza ricorrere a padding aggiuntivo.

La dimensione dello **stride** è stata fissata a $16 \times 16 \times 8$. Questa misura risulta essere un buon compromesso tra la necessità di estrarre un numero sufficiente di campioni per il training e la gestione dei tempi di calcolo e della memoria disponibile. Con uno *stride* più ridotto, infatti, il numero di blocchi sarebbe aumentato in maniera significativa, con un'espansione eccessiva del dataset e un conseguente incremento del costo computazionale. Con i parametri scelti, invece, ciascun volume genera 21 regioni distinte.

Dataset

Il **dataset** è composto da 500 coppie, costituite da un volume originale e dalla corrispondente ricostruzione con SIRT. Per favorire la capacità di generalizzazione della rete, le ricostruzioni sono state generate utilizzando un numero variabile di iterazioni, compreso tra 25 e 150. In questo modo il dataset include esempi con diversi livelli di qualità, consentendo alla rete di imparare a rimuovere artefatti anche da ricostruzioni non ottimali.

Complessivamente, il numero totale di *patches* utilizzate per il training è di:

$$500 \times 21 = 10500 \text{ campioni} \tag{4.5}$$

Di questi, il 10% è stato destinato al *validation set*, mentre il restante 90% è stato utilizzato per il training vero e proprio.

Batch size

Il termine **batch size** fa riferimento alla dimensione di un singolo *batch*. Nel presente lavoro è stato adottato un valore pari a 8. In questo modo la rete aggiorna i pesi ogni otto sottovolumi, basandosi sull'errore medio calcolato su di essi.

Overfitting

Per evitare fenomeni di overfitting, sono state adottate le seguenti tecniche:

- **Early stopping**: consiste nell'interrompere anticipatamente l'allenamento qualora le prestazioni sul *validation set* non migliorino entro un numero prestabilito di epoche.
- **ReduceLROnPlateau**: riduce automaticamente il valore di **learning rate** quando le performance si stabilizzano. Questo parametro rappresenta la velocità con cui i pesi della rete vengono aggiornati durante l'allenamento. Una sua riduzione mirata, in prossimità di un minimo locale della funzione di perdita, consente una convergenza più stabile e accurata.

Nel processo di training utilizzato, il valore di *patience* è stato impostato a 10 per *early stopping* e a 5 per *ReduceLROnPlateau*, con un fattore di riduzione del *learning rate* pari a 0.5.

Funzione di loss

Per l'addestramento del modello sono state testate diverse combinazioni di funzioni di loss, introducendo opportuni fattori di scala per portare le metriche sulla stessa grandezza numerica. Le percentuali di ponderazione sono state definite a seguito di numerosi test sperimentali, con l'obiettivo di individuare la ripartizione più efficace per ogni combinazione di loss. Le configurazioni adottate sono le seguenti:

- **MSE + SSIM**, costruita assegnando un peso del 60% a MSE e del 40% a SSIM, ridimensionando quest'ultimo con un fattore pari a 0.01;
- **L1 + SSIM**, attribuendo un peso del 70% a L1 e del 30% a SSIM, scalando quest'ultimo con un fattore 0.1;
- **L1 + MSE**, assegnando il 65% di peso a L1 e il 35% a MSE, senza introdurre fattori di scala aggiuntivi.

4.3.2 Valutazione

In questa sezione si presentano i risultati relativi al processo di rimozione degli artefatti.

Poiché il tempo di *post-processing* (di circa 4 secondi per volume) si mantiene costante al variare della qualità della ricostruzione iniziale e della funzione di loss, tale valore non verrà analizzato nel dettaglio, in quanto non fornisce informazioni aggiuntive per la valutazione complessiva delle diverse configurazioni. Questo comportamento è dovuto alla natura fissa dell'architettura della U-Net, che esegue un numero invariato di operazioni per ogni volume.

Ciascuno dei tre modelli, corrispondenti alle diverse configurazioni di loss considerate, è stato addestrato per un totale di 50 epoche. Lo *stride* utilizzato durante la fase di test del modello è di dimensione $2 \times 2 \times 2$, sensibilmente più piccolo rispetto a quello impiegato in fase di training ($16 \times 16 \times 8$). Tale differenza riflette le diverse funzioni che svolge nelle due fasi.

In fase di training, lo *stride* ha principalmente lo scopo di generare il dataset, suddividendo i volumi in *patch* di dimensioni compatibili con i requisiti dell'allenamento, mentre durante la fase di *testing*, esso viene ridotto per ottenere un maggior numero di *patch* sovrapposte. Questa scelta favorisce una ricostruzione più stabile e continua, riducendo il rischio di artefatti dovuti a una suddivisione troppo grossolana del volume e evitando che le tracce delle singole *patch* siano visibili nel volume finale post-processato.

Loss combinata MSE + SSIM

Iterazioni iniziali	PSNR	L1	SSIM
25	26.40	0.0226	0.9411
50	28.84	0.0159	0.9491
75	30.69	0.0124	0.9634
100	32.00	0.0102	0.9717
125	32.25	0.0101	0.9757
150	32.26	0.0101	0.9781

Tabella 4.2: Valori medi delle metriche di qualità calcolati sui volumi post-processati mediante la loss combinata MSE + SSIM.

Loss combinata L1 + SSIM

Iterazioni iniziali	PSNR	L1	SSIM
25	24.71	0.0259	0.9260
50	26.93	0.0184	0.9372
75	28.59	0.0143	0.9534
100	29.61	0.0120	0.9624
125	29.91	0.0117	0.9673
150	29.94	0.0118	0.9702

Tabella 4.3: Valori medi delle metriche di qualità calcolati sui volumi post-processati mediante la loss combinata L1 + SSIM.

Loss combinata L1 + MSE

Iterazioni iniziali	PSNR	L1	SSIM
25	25.57	0.0228	0.9340
50	27.77	0.0159	0.9456
75	29.51	0.0123	0.9587
100	30.74	0.0099	0.9670
125	31.16	0.0095	0.9716
150	31.23	0.0093	0.9744

Tabella 4.4: Valori medi delle metriche di qualità calcolati sui volumi post-processati mediante la loss combinata L1 + MSE.

I risultati riportati nelle tabelle 4.2, 4.3 e 4.4, mostrano un miglioramento delle performance della Residual U-Net in relazione al numero di iterazioni SIRT impiegate nella ricostruzione iniziale con ASTRA. Su campioni ottenuti con un numero ridotto di iterazioni (ad esempio 25), la rimozione degli artefatti risulta meno efficace, producendo volumi post-processati di qualità significativamente inferiore. Al contrario, con un numero elevato di iterazioni (fino a 150), la rete è in grado di preservare con maggiore accuratezza le strutture rilevanti e di ricostruirle in modo più fedele, migliorando la qualità complessiva del risultato. Questo comportamento, comune a tutte e tre le configurazioni di loss, dimostra che una ricostruzione iniziale di alta qualità è un prerequisito fondamentale per un post-processing efficace.

A partire da queste osservazioni, è possibile analizzare in dettaglio le differenze prestazionali tra le tre combinazioni di loss considerate, confrontando le metriche di errore per identificare quale configurazione consenta di ottenere i risultati migliori.

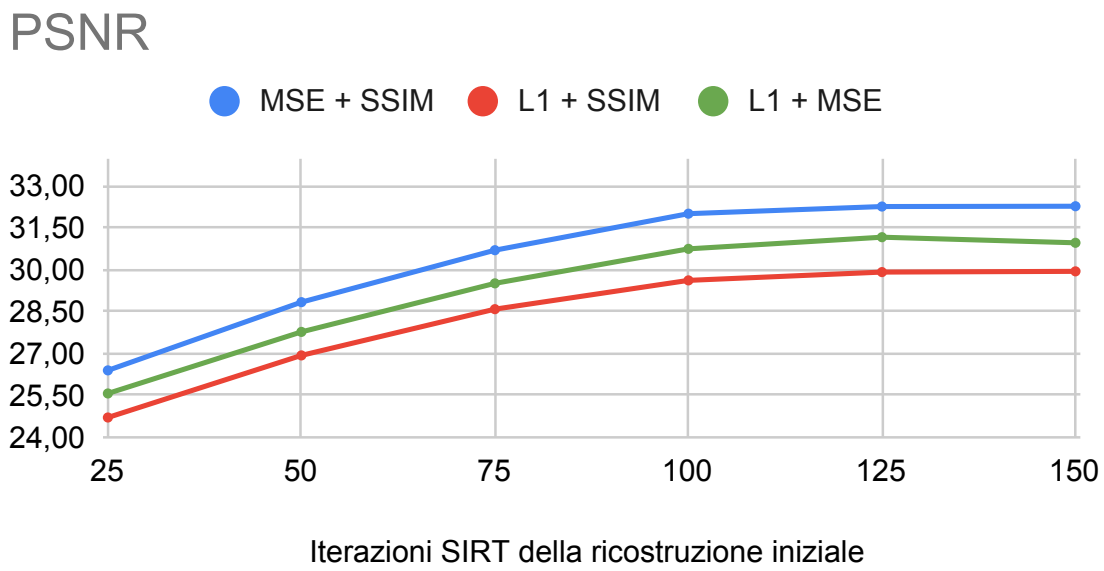


Figura 4.3: Confronto dei valori di PSNR, calcolati sui volumi post-processati, in funzione della combinazione di loss adottata e del numero di iterazioni SIRT impiegate nella ricostruzione iniziale.

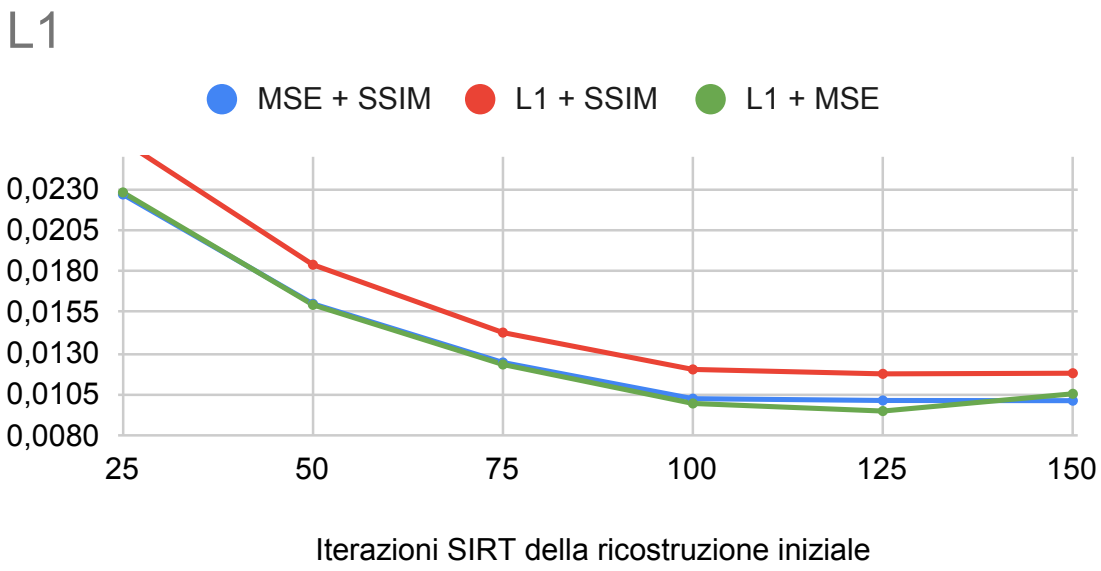


Figura 4.4: Confronto dei valori di L1, calcolati sui volumi post-processati, in funzione della combinazione di loss adottata e del numero di iterazioni SIRT impiegate nella ricostruzione iniziale.

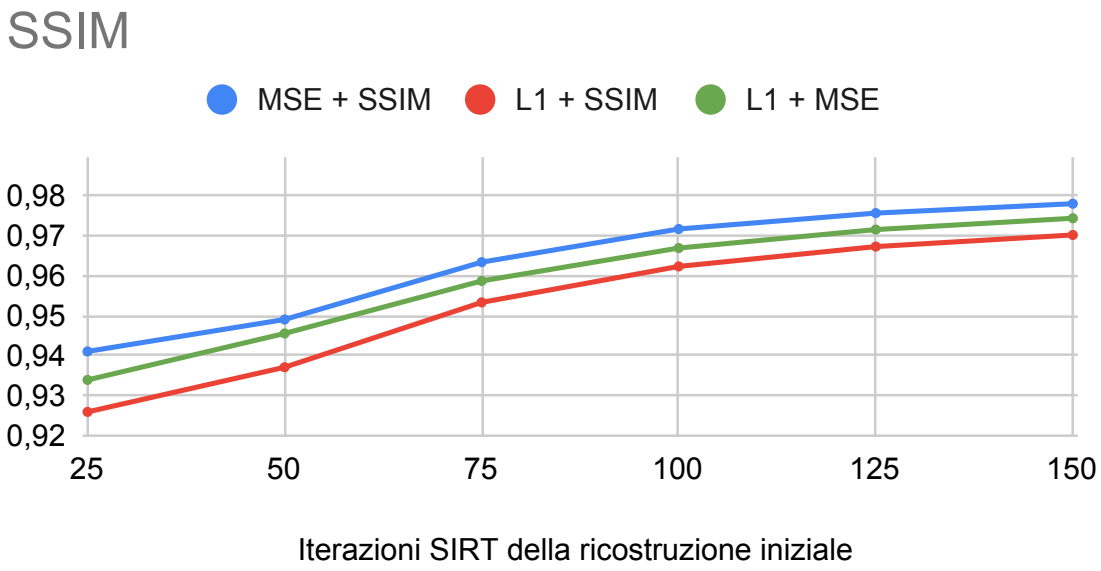
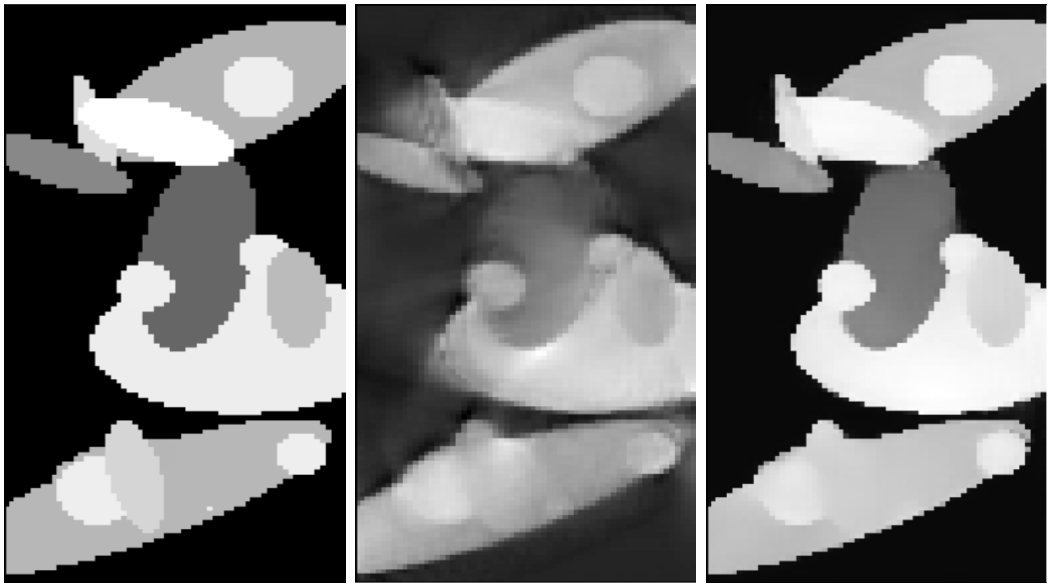
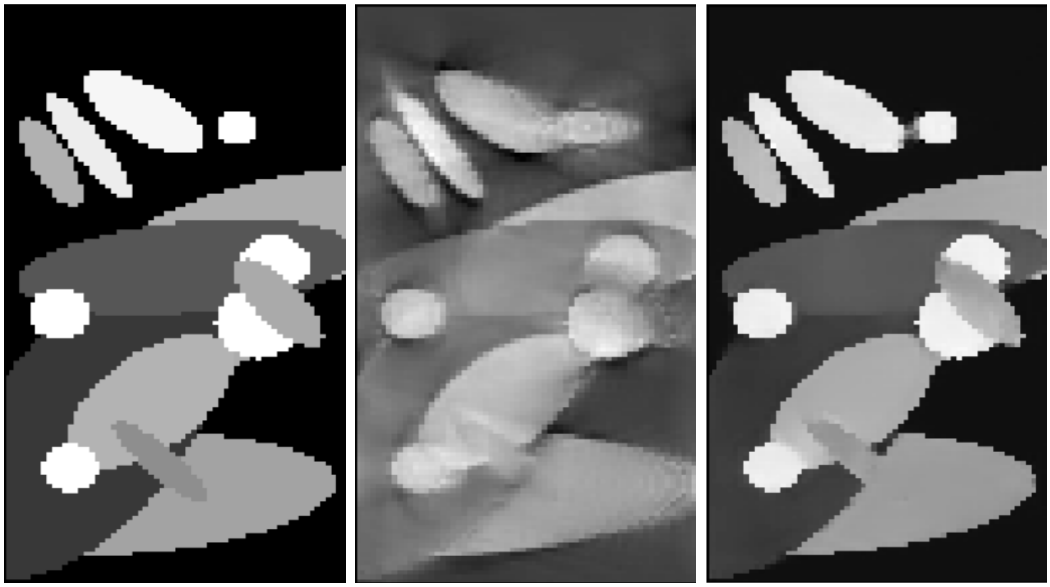


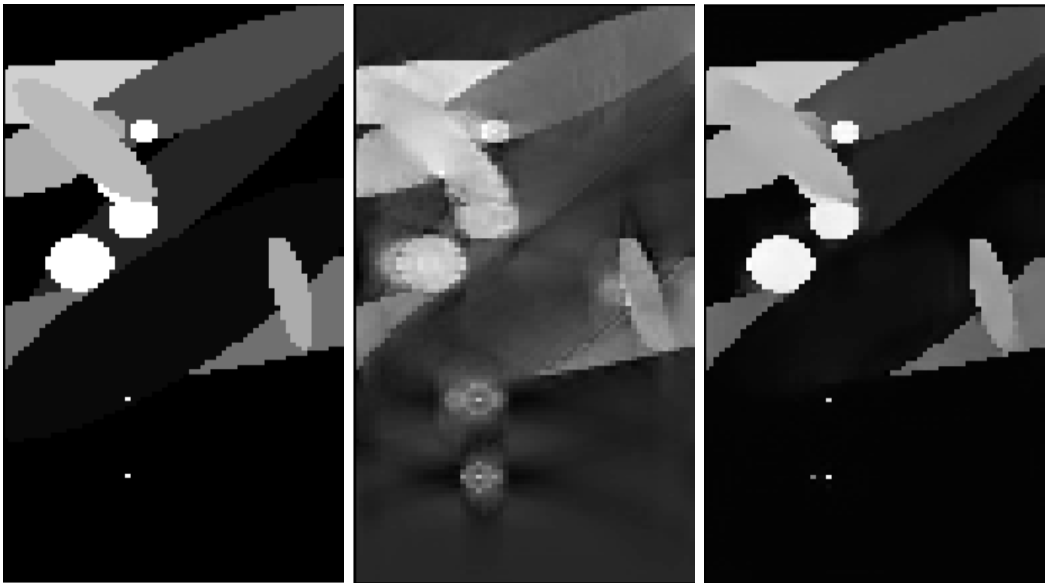
Figura 4.5: Confronto dei valori di SSIM, calcolati sui volumi post-processati, in funzione della combinazione di loss adottata e del numero di iterazioni SIRT impiegate nella ricostruzione iniziale.



(a) Ricostruzione post-processata con 25 iterazioni SIRT e funzione di loss L1 + MSE. Il basso numero di iterazioni comporta una ricostruzione meno accurata dei bordi, causando una perdita di dettaglio nelle regioni con figure sovrapposte di densità simile. Da sinistra a destra sono visualizzati: immagine originale, ricostruzione iniziale e volume finale post-processato.



(b) Ricostruzione post-processata con 75 iterazioni SIRT e funzione di loss L1 + SSIM. La definizione dei bordi è accurata, grazie al numero relativamente alto di iterazioni iniziali. Da sinistra a destra sono visualizzati: immagine originale, ricostruzione iniziale e volume finale post-processato.



(c) Ricostruzione post-processata con 125 iterazioni SIRT e loss MSE + SSIM. L'alto numero di iterazioni e la loss SSIM permettono di ricostruire anche dettagli molto piccoli (punti bianchi), sebbene possano comparire artefatti puntuali vicino a picchi intensi (vedi immagine a destra). Da sinistra a destra sono visualizzati: immagine originale, ricostruzione iniziale e volume finale post-processato

Dall'analisi comparativa delle tre funzioni di loss emergono differenze significative nei risultati ottenuti.

La combinazione MSE + SSIM si dimostra la più efficace: i volumi post-processati raggiungono valori più alti sia in termini di PSNR (≈ 32.2) che di SSIM (≈ 0.98). Questo risultato indica che l'abbinamento di un criterio numerico (MSE) con uno percettivo (SSIM) favorisce la preservazione delle strutture fini e la riduzione degli artefatti.

Al contrario, L1 + SSIM, sebbene mantenga un abbinamento di criteri simile a quello precedente, registra le performance più deboli (PSNR ≈ 29.9, SSIM ≈ 0.97), evidenziando una minore efficacia della penalizzazione L1 rispetto a MSE.

In una posizione intermedia si colloca L1 + MSE, che ottiene metriche migliori rispetto alla loss precedente (PSNR ≈ 31.2, SSIM ≈ 0.97), e leggermente inferiori rispetto alla prima combinazione. Si osserva, inoltre, un errore L1 identico a quello di MSE + SSIM (Fig. 4.4), indicando una simile accuratezza numerica; l'assenza della componente SSIM non consente, tuttavia, di raggiungere la stessa qualità percettiva.

In conclusione, la loss combinata MSE + SSIM, bilanciando accuratezza numerica e fedeltà percettiva, rappresenta la scelta ottimale per la fase di *post-processing*.

Integrando i risultati presentati nella Sez. 4.2.2 con quelli ottenuti mediante la loss MSE + SSIM, emerge come il numero ideale di iterazioni SIRT sia intorno alle 100. Oltre questa soglia, infatti, non si osservano miglioramenti significativi nella fase di *post-processing*, rendendo superfluo un ulteriore aumento della qualità della ricostruzione iniziale. Questo approccio consente di ridurre il tempo di elaborazione totale a circa 15 s per volume (10.8 s per la ricostruzione SIRT e 4 s per il *post-processing* mediante Residual U-Net), mantenendo una ricostruzione finale di alta qualità, adatta per l'identificazione di eventuali oggetti al suo interno.

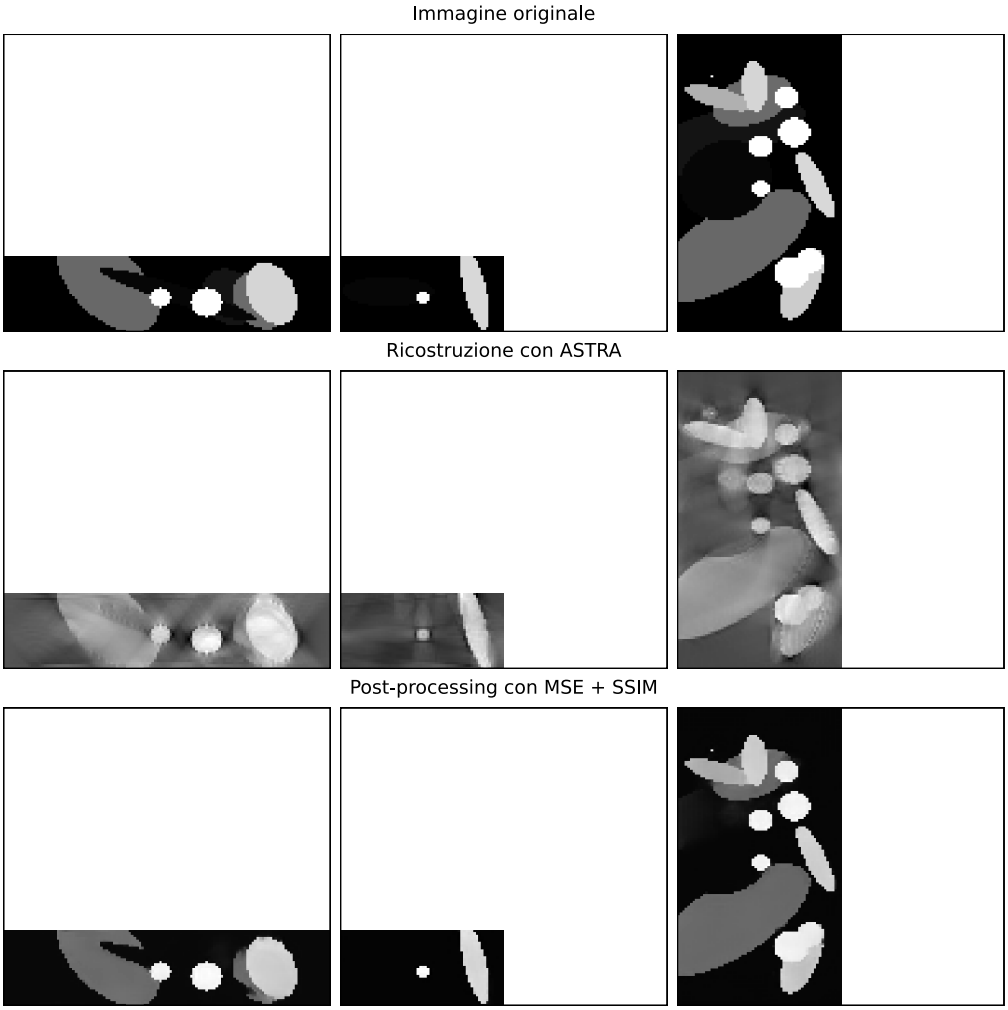


Figura 4.8: Esempio di ricostruzione post-processata ottenuta con i parametri ottimali individuati (100 iterazioni SIRT e loss MSE + SSIM). Si evidenzia una netta riduzione degli artefatti.

Conclusioni

Il presente lavoro ha indagato l'efficacia della tomografia traslazionale nello screening dei bagagli, proponendo un approccio innovativo che integra ricostruzione iterativa e modelli di deep learning per la riduzione degli artefatti.

Attraverso l'impiego della libreria ASTRA Toolbox, è stata sviluppata una geometria di acquisizione originale, in grado di riprodurre il movimento traslazionale di un bagaglio sul nastro trasportatore e di acquisire le proiezioni del volume mediante due sorgenti posizionate in configurazione verticale e laterale. Questa geometria ha consentito di simulare un contesto operativo realistico, dimostrando il potenziale della tomografia traslazionale come alternativa valida alla tomografia rotazionale, soprattutto in scenari caratterizzati da vincoli meccanici e/o operativi.

La fase iniziale di ricostruzione è stata svolta con l'algoritmo iterativo SIRT, il quale ha mostrato robustezza anche in presenza di dati limitati, tipici della tomografia traslazionale. L'analisi dei risultati ha evidenziato come una frequenza di emissione dei raggi pari a 50 Hz sia sufficiente per ottenere ricostruzioni di buona qualità. In seguito alla fase di *post-processing*, svolta mediante il modello di Residual U-Net, si è anche stabilito che 100 iterazioni SIRT rappresentano un compromesso ideale tra qualità e carico computazionale, garantendo valori di SSIM elevati senza oneri eccessivi.

L'approccio di *post-processing* basato su deep learning, implementato con una funzione di loss che combina MSE e SSIM, ha prodotto ricostruzioni finali con valori di PSNR superiori a 32 dB e SSIM superiori a 0.97, confermando l'efficacia della strategia proposta.

Sono emerse tuttavia alcune limitazioni, legate principalmente alle risorse computazionali disponibili. L'elaborazione è stata condotta su una GPU T4 che, pur garantendo risultati promettenti, non eguaglia le prestazioni dei moderni sistemi CT. Il tempo medio di ricostruzione della configurazione ottimale, pari a circa 15 secondi per volume, sebbene accettabile in fase sperimentale, risulterebbe insufficiente

in scenari operativi real-time, dove sarebbe necessario implementare l'approccio su infrastrutture di calcolo più performanti.

Le prospettive future prevedono l'adozione di geometrie di acquisizione più realistiche, l'utilizzo di dataset reali per l'addestramento e lo sviluppo di architetture di reti neurali più avanzate o ibride, finalizzate a migliorare ulteriormente la rimozione degli artefatti.

I risultati ottenuti dimostrano la validità dell'approccio basato sulla tomografia traslazionale e pongono solide basi per lo sviluppo di sistemi di screening più affidabili, volti a incrementare sicurezza ed efficienza nei contesti aeroportuali.

Bibliografia

[1] Avinash C. Kak e Malcolm Slaney. *Principles of Computerized Tomographic Imaging*. Reprint of the 1988 edition. Philadelphia: Society for Industrial e Applied Mathematics, 2001. ISBN: 9780898714944.

[2] L. S. Shepp e B. F. Logan. «The Fourier reconstruction of a head section». In: *IEEE Transactions on Nuclear Science* NS-21 (1974).

[3] Gregory Mogilevsky et al. «Raman Spectroscopy for Homeland Security Applications». In: *International Journal of Spectroscopy* 2012 (2012), pp. 1–11. URL: https://www.researchgate.net/publication/258387262_Raman_Spectroscopy_for_Homeland_Security_Applications.

[4] Peter Hanson. *How Are CT Scanners Changing Airport Security?* 2024. URL: https://simpleflying.com/ct-scanners-airport-security-answer/.

[5] Jens Gregor e Thomas Benson. «Computational Analysis and Improvement of SIRT». In: *IEEE Transactions on Medical Imaging* 27.7 (2008), pp. 918–924. DOI: 10.1109/TMI.2008.923696.

[6] Alex Krizhevsky, Ilya Sutskever e Geoffrey E. Hinton. «Imagenet classification with deep convolutional neural networks». In: *Advances in Neural Information Processing Systems (NIPS)*. 2012, pp. 1106–1114.

[7] Simon J.D. Prince. *Understanding Deep Learning*. The MIT Press, 2023.

[8] Olaf Ronneberger, Philipp Fischer e Thomas Brox. «U-Net: Convolutional Networks for Biomedical Image Segmentation». In: *arXiv preprint arXiv:1505.04597* (2015). Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany.

[9] Md Zahangir Alom et al. «Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation». In: *arXiv preprint arXiv:1802.06955* (2018).

[10] Luis Perez e Jason Wang. *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. 2017. arXiv: 1712.04621 [cs.CV]. URL: https://arxiv.org/abs/1712.04621.

Ringraziamenti

I ringraziamenti sono probabilmente la parte più difficile da scrivere dell'intera tesi. Trovare le parole giuste per ringraziare tutte le persone importanti della mia vita, senza scadere nel banale o nello scontato, non è affatto semplice e temo che anche io finirò per ricadere un po' in questo rischio.

Un ringraziamento speciale va sicuramente alla prof.ssa Piccolomini, per l'enorme disponibilità dimostrata in questi mesi e per la gentilezza con cui si è sempre rivolta a me.

Ringrazio anche il mio co-relatore Davide Evangelista, che grazie alla sua conoscenza e guida ha reso possibile lo sviluppo di questa tesi.

Subito dopo voglio ringraziare i miei genitori, veri promotori di questo percorso. Grazie per tutte le opportunità che mi avete dato, per aver sempre creduto in me e per gli insegnamenti che mi hanno reso la persona che sono oggi. Vi devo tutto e vi voglio tanto bene.

Un grazie davvero grande va a mio fratello Mattia, che è sempre stato per me un esempio da seguire. Mi ha dato tanto, forse senza neanche rendersene conto. Spero di riuscire a passare più tempo con te in futuro.

Ringrazio Hillary per il sostegno, la mia nipotina Ginevra, che riesce sempre a strapparmi un sorriso, e i miei nonni — anche quelli che ormai mi guardano da lassù — a cui non smetterò mai di voler bene. Vi porterò sempre nel cuore.

Un ringraziamento speciale va a Beatrice, la mia ragazza, per tutti i momenti preziosi trascorsi insieme, per avermi sempre incoraggiato nei momenti difficili e per l'amore che ha portato nella mia vita. Sei una persona speciale e resterai sempre nel mio cuore.

Grazie anche a Cino, Chiara, Bruni, Tao, Li, Sam e Mazz per i momenti di svago condivisi durante questi tre anni di università, e ai miei coinquilini Marco e Lorenzo: compagni di bevute, film, scherzi e carbonare.

Ringrazio tutti i miei amici e in particolare Gabri, Fabio, Dandra, Marco e Gianluca, per tutti i momenti condivisi insieme: le risate, gli insulti, i McDonald alle 2 di notte, i viaggi in macchina alla "Una notte da leoni", le discussioni profonde e quelle meno profonde. Vi voglio bene ragazzi.

Infine, voglio ringraziare tutti gli imprevisti e le sfide affrontate in questi ultimi anni. Sono quelle che mi hanno fatto crescere maggiormente, parti integranti di una vita piena di momenti difficili, ma anche di esperienze indimenticabili.

Grazie a tutti voi che mi avete reso la persona che sono oggi.