Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica per il management

RETI NEURALI CONVOLUZIONALI PER IL MIGLIORAMENTO DI IMMAGINI TOMOGRAFICHE AD ANGOLI LIMITATI

Relatore:
Chiar.mo Prof.ssa
Elena Loli Piccolomini

Correlatore:
Chiar.mo Dott.
Davide Evangelista

Presentata da:
Fabian Vincenzi

II Sessione
Anno Accademico 2020/2021

Indice

Introduzione

Nel corso della tesi verrà presentata una rete neurale convoluzionale e un suo utilizzo per la ricostruzione di immagini CT (Computed Tomography) a bassa dose. Le reti neurali convoluzionali fanno parte del deep learning, cioè algoritmi ispirati alla struttura e alla funzione del cervello. In particolare, le reti neurali convoluzionali riescono a riconoscere cosa un'immagine rappresenta. Nel nostro caso utilizzeremo una rete in grado di ricostruire delle immagini corrotte, infatti queste immagini saranno ricavate dalla CT a bassa dose. La CT è una tecnologia per effettuare scansioni su parti del corpo umano che poi sono restituite sotto forma di immagine, poichè durante il procedimento vengono emesse radiazione e quello completo dura molto tempo, se ne è sviluppato uno veloce che dura poco tempo e permette di avere una bassa esposizione alle radiazioni. Il problema della CT a bassa dose è che l'immagine risultato perde di qualità e serve applicare degli algoritmi per migliorarla. Sono presenti degli algoritmi di ottimizzazione per questo problema, ma essi potrebbero produrre artefatti. Quindi per risolvere questo problema vengono usate delle tecniche di post-processing.
In questa tesi verranno affrontati nei vari capitoli tutti questi argomenti, a partire dal Capitolo 1 con una breve introduzione al Deep Learning e a seguire un approfondimento su un loro possibile utilizzo, le reti neurali convoluzionali. Di quest'ultime verrà spiegato il loro funzionamento e la loro architettura. Il Capitolo 2 tratterà della Computed Tomography (CT), come si è evoluta negli anni e la trasformata di Radon e la sua inversa. Nel Capitolo 3 è presentata una rete neurale convoluzionale encoder-decoder residua per ct a basso dosaggio, che sarà utilizzata nel capitolo successivo. Infatti, nel Capitolo 4 saranno riportate tutte le varie prove effettuate per avere la rete il più efficace possibile.

Capitolo 1

Reti neurali convoluzionali

1.1 Introduzione al deep learning

Il Deep Learning (DL), letteralmente apprendimento approfondito, è la sottocategoria del Machine Learning che fa riferimento agli algoritmi ispirati alla struttura e alla funzione del cervello.
Il DL non è un algoritmo per l'esecuzione di task specifici, ma si basa su algoritmi per l'assimilazione di dati.
Questi algoritmi sono composti da vari livelli non lineari a cascata, ciascun livello utilizza l'output del livello precedente come input, così facendo si riescono ad estrapolare caratteristiche di più alto livello partendo da quelle di più basso livello. Il DL tramite dei sistemi artificiali simula i processi di apprendimento del cervello biologico per insegnare alle macchine non solo ad apprendere autonomamente, ma a farlo come sa fare il cervello umano.
Se per esempio vogliamo usare la rete neurale per il riconoscimento visivo, il primo livello potrebbe imparare a riconoscere i bordi, il secondo a riconoscere forme più complessi, come triangoli e rettangoli e via via aumentando aumentano i dettagli che vengono riconosciuti. Grazie ai molteplici livelli di astrazione le reti neurali possono imparare meglio a risolvere complessi problemi di riconoscimento di schemi poiché ad ogni livello aggiungono informazioni e analisi utili per l'output.
La qualità del risultato quindi dipende da quanti livelli sono presenti nella rete, mentre la scalabilità di essa dipende dai data set, dai modelli matematici e dalle risorse computazionali, infatti i sistemi di DL migliorano le proprie prestazioni all'aumentare dei dati disponibili.
Le architetture di DL vengono sempre più utilizzate e sono per esempio applicate nel riconoscimento della lingua parlata, nel riconoscimento audio, nella bioinformatica e nella computer vision, tra cui in quello che sarà oggetto dei prossimi capitoli, le reti neurali convoluzionali (CNN).

1.2 Introduzione alle reti neurali convoluzionali

Le reti neurali convoluzionali o Convnet (CNN dall'inglese convolutional neural network) fanno parte della computer vision e sono uno degli algoritmi di Deep Learning più utilizzati in questo ambito. Essi trovano applicazione in tantissimi campi tra cui le automobili autonome, i droni e le diagnosi mediche.
L'applicazione più popolare di una rete neurale convoluzionale resta quella di identificare cosa un'immagine rappresenta, infatti tramite una CNN il computer è in grado di classificare cosa un'immagine mostra e identificare con buona probabilità il suo contenuto.
Le reti neurali sono costruite per analizzare immagini incluse dentro certi set di dati e classificare gli oggetti nelle immagini al loro interno, per esempio non puoi ottenere un riscontro positivo se fai analizzare un'immagine di un volto umano a una CNN che accetta solo immagini di autoveicoli, poiché non capirebbe le forme e gli oggetti che l'immagine rappresenta.
Le reti neurali convoluzionali, come tutte le reti neurali, hanno un layer di input, uno o più layer nascosti, che effettuano calcoli tramite funzioni di attivazione, e un layer di output con il risultato, la differenza è appunto le convoluzioni.
Un esempio di architettura di rete neurale convoluzionale può essere il seguente:

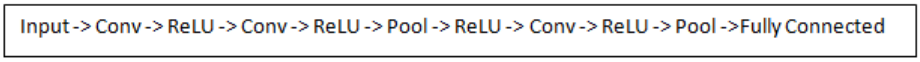


Figura 1.1: Rappresenta i vari livelli di una rete neurale convoluzionale.

Nel quale Input, Conv, ReLU, Pool e Fully Connected identificano le tre tipologie di layer della rete neurale convoluzionale.

Quindi abbiamo:

- **Livello di input:** rappresenta l'insieme di numeri che costituisce l'immagine da analizzare per il computer. Ad esempio, 32 x 32 x 3 indica larghezza (32), altezza (32) e profondità (3, i colori del formato RGB) dell'immagine.

- **Livello convoluzionale:** è il livello principale della rete. Il suo obiettivo è quello di individuare schemi, come ad esempio curve, angoli e quadrati. Sono più di uno e ognuno di essi si concentra nella ricerca di queste caratteristiche nell'immagine iniziale. Maggiore è il numero e maggiore è la complessità della caratteristica che riescono ad individuare.

- **Livello ReLu (Rectified Linear Units):** si pone l'obiettivo di annulare valori negativi ottenuti nei livelli precedenti.
- **Livello Pool:** permette di identificare se la caratteristica di studio è presente nel livello precedente.
- **Livello FC (o Fully connected, completamente connesso):** connette tutti i neuroni del livello precedente al fine di stabilire le varie classi identificative secondo una determinata probabilità. Ogni classe rappresenta una possibile risposta finale.

1.3 Livello convoluzionale

La parola "convoluzione" significa "far scorrere" una funzione(blu) sopra un'altra (rossa), di fatto "mescolandole" insieme ottenendo una funzione (verde) che di fatto rappresenta il prodotto delle due funzioni.

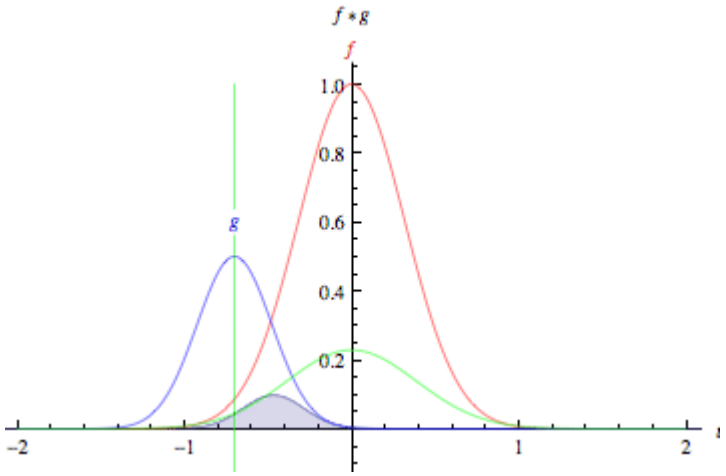


Figura 1.2: L'immagine è un esempio di convoluzione, la funzione rossa rappresenta la funzione di base, quella blu è la funzione che scorre sopra la funzione di base e quella verde è il risultato ottenuto.

In questo caso la funzione rossa rappresenta l'immagine in input, mentre quella blu è conosciuta come "filtro", perché identifica una particolare caratteristica dell'immagine che si vuole identificare.
Per i primi livelli il filtro rappresenta una caratteristica di basso livello perché identifica semplici oggetti come curve e linee. Per un livello si identificheranno le curve, poi in un altro le linee orizzontali, per un altro le circonferenze fino a formare figure complesse che rappresentano oggetti più complicati. In quest'ultimo caso si dice che il filtro rappresenta una caratteristica di alto livello, poiché identifica oggetti complessi, come una mano o un volto.
Una volta stabilito cosa il filtro deve identificare si decide la dimensione del filtro e il numero di filtri da utilizzare nel livello.
Ipotizziamo due filtri, uno per le linee verticali e uno per quelle orizzontali, dalle dimensioni 2 x 2. Successivamente si parte ad analizzare il campo ricettivo, che ha la stessa dimensione del filtro, ed è inizialmente rappresentato dal primo blocco di pixel 2 x 2 in alto a sinistra.
Il risultato si ottiene facendo un prodotto scalare dei valori del filtro con i valori di questo primo blocco.
Nel caso in cui il risultato non è in prossimità di linee esso assumerà valore 0.
Questa operazione va ripetuta per tutti i blocchi che l'immagine di input

può contenere, di conseguenza il campo ricettivo viene fatto spostare ogni volta di un determinato passo verso destra (in questo caso 1).

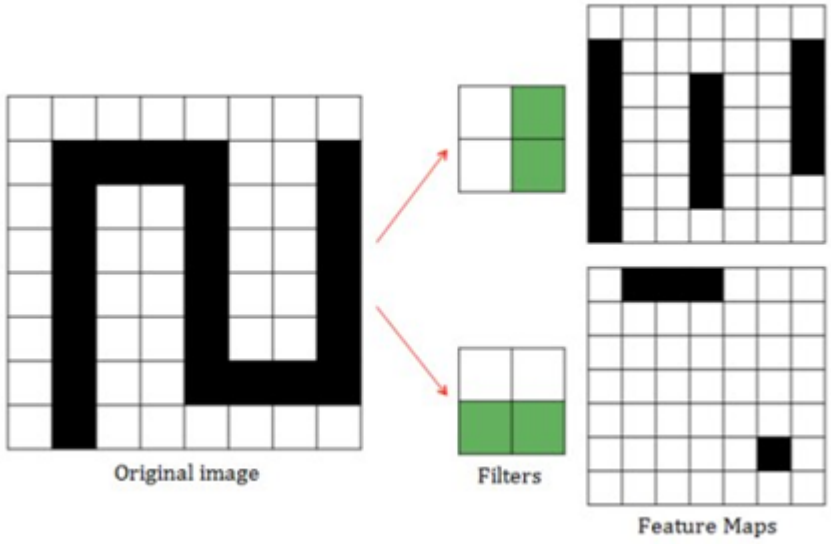


Figura 1.3: A sinistra possiamo vedere l'immagine di input, le due frecce indicano due percorsi possibili che hanno un filtro diverso e l'immagine finale è il risultato del filtro applicato all'immagine iniziale.

Dopo aver fatto scivolare il campo ricettivo su tutte le posizioni, otteniamo una matrice 7 x 7 x 2.
Il 2 rappresenta la profondità che sarebbe il numero di filtri usati.
L'insieme dei valori che si ottengono seguendo questa procedura si dice mappa di attivazione.
Ci sono 4 parametri principali (definiti iperparametri) che influenzano il comportamento di un livello convoluzionale (passo, dimensione del filtro, numero e riempimento zero).
Il riempimento zero, o zero-padding dall'inglese, identifica uno strato da apporre al volume di input iniziale per non perdere alcune informazioni nel passaggio da un livello ad un altro.
Se per esempio volessimo applicare un filtro 5 x 5 x 3 ad un volume di input 32 x 32 x 3, il volume di output sarebbe 28 x 28 x 3. Si può notare che le dimensioni diminuiscono e continuando ad applicare i livelli convoluzionali la dimensione continuerà a diminuire.
Nei primi strati della rete è però utile conservare il maggior numero di informazioni sul volume di input per estrarre caratteristiche di basso livello che altrimenti andrebbero perse e sarebbe impossibile recuperarle nei livelli successivi.
Per far ciò è possibile applicare uno spessore zero di dimensione 2 al primo li-

vello, quindi si riempie il volume di input con zeri attorno al bordo facendole diventare 36 x 36 x 3.

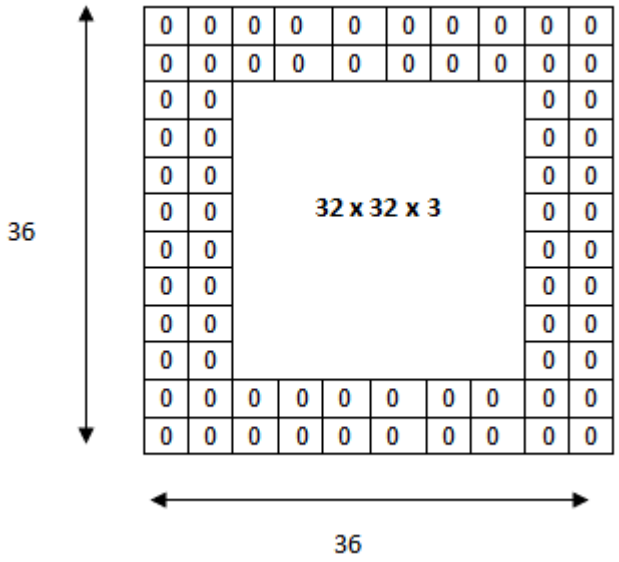


Figura 1.4: Questo è un esempio di zeropadding, ovvero la tecnica di applicare un bordo, in questo caso di spessore 2, di valori 0, per far sì che l'applicazione di un filtro consideri anche i valori ai bordi dell'immagine originale, ingrandendo di fatto le dimensioni dell'immagine.

1.4 Livello ReLU

Quando passiamo ad un altro livello, l'output del primo livello diventa l'input del secondo livello.
Di conseguenza, l'output del livello convoluzionale diventa l'input del livello successivo, che solitamente è il livello ReLu.
Il livello ReLU rappresenta un livello non lineare, il cui scopo è quello di introdurre la non linearità a un sistema che calcola operazioni lineari durante i livelli convoluzionali.
In questo modo, la rete è in grado di allenarsi molto più velocemente senza impattare significativamente sull'accuratezza dei risultati.
In questo livello vengono sostituiti tutti i valori negative delle feature map con lo zero, aumentando le proprietà non lineari del modello e della rete globale senza influenzare i campi ricettivi del livello convoluzionale.

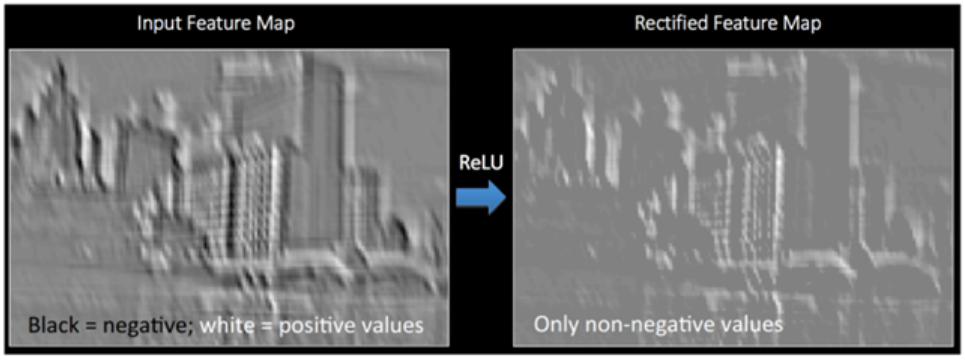


Figura 1.5: A sinistra vediamo l'immagine di input a cui non è stato applicato ancora il Relu che quindi contiene anche valori negativi. A destra vediamo la stessa immagine dopo che è stato applicato il ReLu e che quindi non presenta più valori negativi.

1.5 Livello Pool

Dopo alcuni livelli ReLU, si può applicare un livello Pool.
Questo livello può essere eseguito in diversi modi: Max (il più popolare), Average, Sum etc.
In questo livello viene utilizzato un filtro e un passo della stessa lunghezza. Questo filtro viene applicato al volume di input del livello precedente e, nel caso di Max Pooling, genera il massimo in ogni campo ricettivo attorno al quale il filtro ruota.

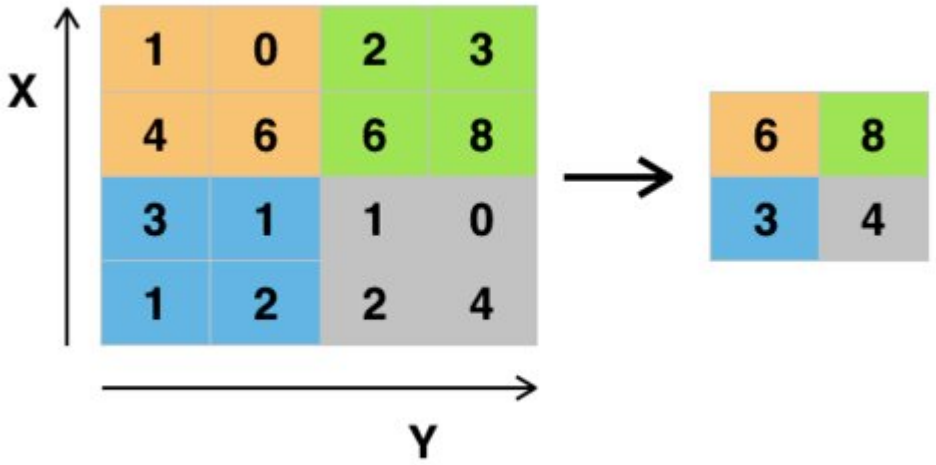


Figura 1.6: In questa immagine viene applicato il max-pooling. L'immagine di sinistra rappresenta i valori dell'immagine in input. Di questa immagine viene preso per ogni quadrato (in questo caso 2 x 2, ognuno colorato diversamente) il valore massimo. A destra abbiamo il risultato dell'intera operazione, infatti sono presenti i 4 valori più alti del loro quadrato corrispondente.

Ad esempio, nel primo riquadro il 6 risulta il massimo e quindi viene riportato nella prima posizione e così via.
Questo livello riduce le dimensioni spaziali del volume di input e i requisiti computazionali per i livelli futuri.

1.6 Livello FC (Fully connected)

Il livello FC (Fully connected) solitamente è l'ultimo di una rete neurale.
Questo livello prende un volume di input e genera un vettore di dimensione N, dove N è il numero di classi tra cui il programma deve scegliere.
Ogni classe sarà associata alla probabilità che nell'immagine sia rappresentata quella classe.
La somma delle probabilità è 1, quindi ogni classe avrà un numero compreso fra 0 e 1.
Ad esempio, in un programma di classificazione delle cifre N sarà 10, poiché le cifre sono 10 (0,1,2,3,4,5,6,7,8,9). Ogni numero rappresenta la probabilità di una certa classe.
Se il vettore risultante per un programma di classificazione di cifre è

$$[0\ 0\ 15\ 10\ 0\ 0\ 0\ 65\ 0\ 10]$$

allora questo rappresenta una probabilità del 15% che l'immagine sia 2, una probabilità del 10% che l'immagine rappresenti un 3, una probabilità del 65% che l'immagine sia un 7 e una probabilità del 10% che l'immagine sia un 9 (tutti gli altri numeri hanno probabilità nulla di essere scelti).
Questo livello completamente connesso funziona poiché guarda l'output del livello precedente e determina quali caratteristiche sono maggiormente correlate a una particolare classe.
Se per esempio, il programma prevede che un'immagine sia un cane, avrà identificato nei livelli precedenti caratteristiche di alto livello come una zampa o il muso.
Per il risultato finale, un livello FC guarda quali caratteristiche di alto livello sono maggiormente correlate ad una particolare classe e calcola i prodotti tra i pesi e il livello precedente per ottenere le probabilità corrette per le diverse classi.

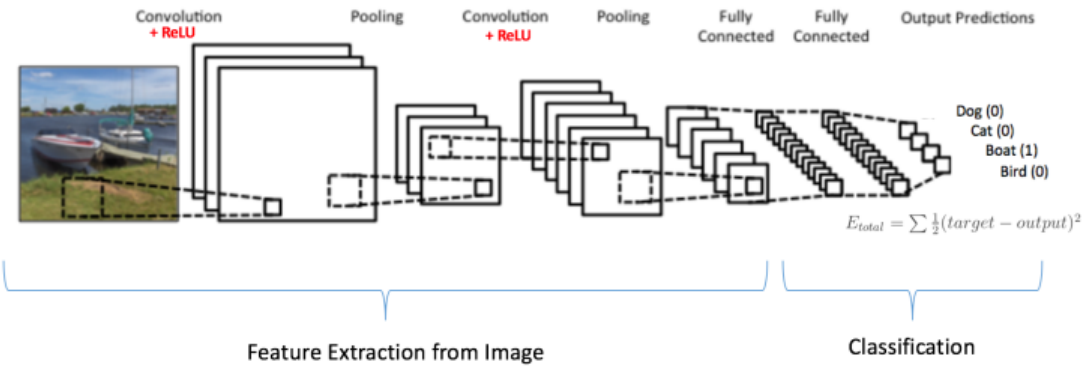


Figura 1.7: Qui possiamo vedere tutti i passaggi della rete a partire dall'immagine iniziale. In particolare alla fine possiamo vedere la parte di FC che si

conclude con l'output delle probabilità che nella immagine sia presente una certa classe.

1.7 Processo di training

Il processo completo di training di una CNN si riassume quindi nei seguenti passi:

- **Step 1:** Inizializziamo tutti i filtri e parametri/pesi con valori random
- **Step 2:** Si prende una immagine di training come input, si svolgono tutti i livelli e si trova l'output delle probabilità finali
- **Step 3:** Si calcola l'errore totale dell'output
- **Step 4:** Si usa la Backpropagation per calcolare il gradiente d'errore rispetto ai pesi della rete e si aggiornano i filtri, i pesi e i parametri per minimizzare l'errore dell'output
- **Step 5:** Si ripetono gli step dal 2 al 4 con tutte le immagini di training

Con questi step si svolge il train della rete che in pratica ottimizza tutti i pesi e i parametri per classificare le immagini. Quando una nuova immagine sarà data in input alla rete, se l'addestramento è stato fatto correttamente, la rete classificherà l'immagine correttamente tra le diverse classi.

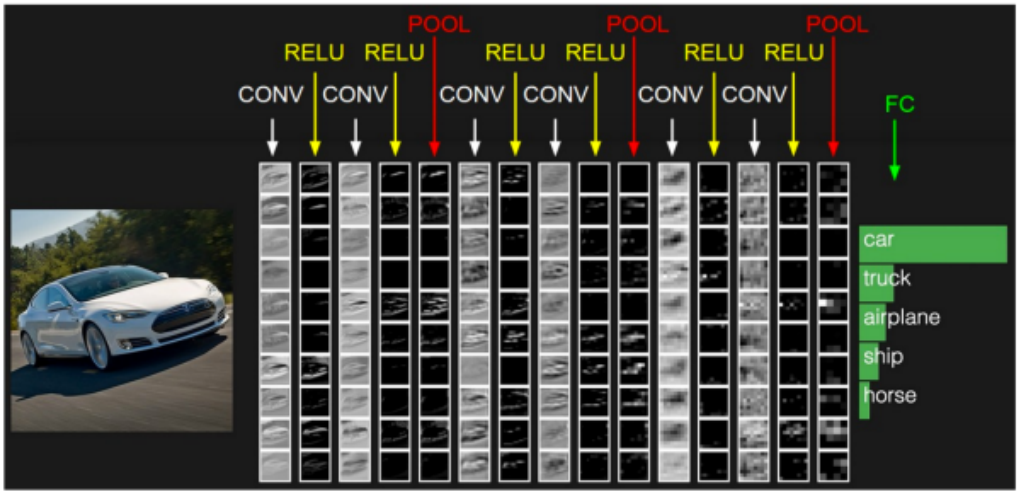
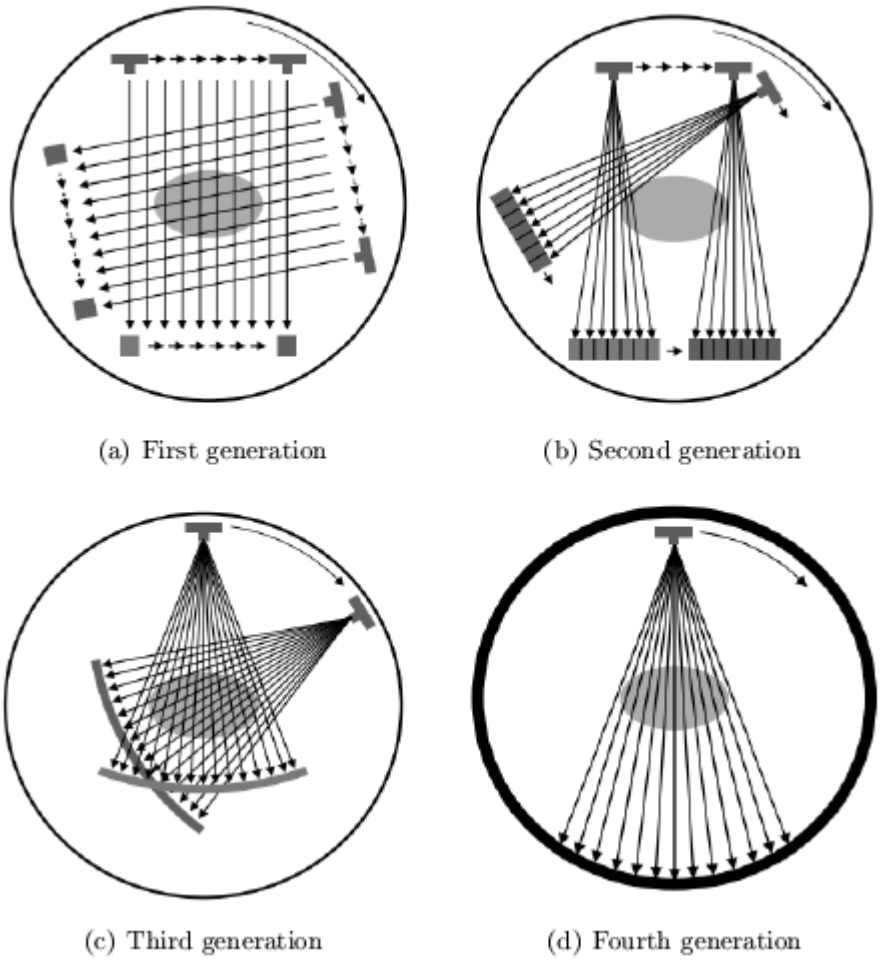


Figura 1.8: Nell'immagine possiamo vedere un esempio di CNN che partendo da una immagine di un'auto svolge tutti i livelli e arriva ad una risposta.

Capitolo 2

CT (Computed Tomography)

La nascita negli anni '70 della tomografia computerizzata (CT, Computed Tomography) ha rivoluzionato la diagnostica delle immagini. Per eseguire la CT classica bisogna acquisire immagini da molti angoli, in una traiettoria circolare. Questa tecnica ha fornito alcuni importanti vantaggi rispetto alla radiografia convenzionale, tra cui la localizzazione in profondità.



(a) First generation (b) Second generation

(c) Third generation (d) Fourth generation

Figura 2.1: L'immagine rappresenta le evoluzioni nel tempo della tomografia, dalla tecnologia primordiale di Cormack e Hounsfield, fino alla soluzione più moderna.

Il primo macchinario fu progettato dal fisico Allan Cormack e dall'ingegnere Godfrey Hounsfield e fu installato a Londra nel 1971; questa rivoluzionaria invenzione fece vincere il premio Nobel per la Medicina ai due inventori.
Il primo prototipo sfruttava una tecnologia basilare, veniva emesso un solo fascio a matita dalla sorgente di raggi X e veniva rilevato da un sensore posto dietro all'oggetto da scansionare, successivamente la coppia sorgente-rilevatore doveva convertirlo in modo da poter proiettare l'intero oggetto, come mostrato nella figura 2.1(a). Dopo questo, si roteava la coppia sorgente-rilevatore per effettuare altre proiezioni complete fino a completare un giro di 180 gradi. L'angolo di rotazione tra due proiezioni successive era molto

piccolo, quindi l'intero processo di scansione durava fino a 4-5 minuti, mentre la prima ricostruzione bidimensionale di 13 mm di una fetta di cervello richiedeva circa 9 giorni.
La seconda generazione di CT (figura 2.1(b)) sfruttava un fascio a ventaglio largo 3-20 gradi, la cui attenuazione viene misurata su più rilevatori contemporaneamente. Prima di ruotare la coppia sorgente-rilevatore è comunque necessaria la fase di conversione per coprire tutto l'oggetto. Con questo metodo una scansione richiedeva 15-30 secondi, tuttavia il campo di misura era ancora piccolo e a causa dei lunghi tempi di acquisizione queste due generazioni erano limitate all'imaging del cranio.
L'obiettivo principale degli sviluppi successivi era di ridurre il tempo di acquisizione a meno di 20 secondi, in modo che fosse possibile acquisire l'immagine di un addome con il minor errore di movimento mentre il paziente trattiene il respiro. Negli anni 90, uscì una terza generazione che utilizzava un fascio a ventaglio largo 40-60 gradi che poteva scansionare tutto l'oggetto in un singolo colpo (figura 2.1(c)). Ovviamente è stato installato un rilevatore più grande, con fino a 800-1000 celle di registrazione. In questo modello non è richiesta la fase di conversione e lo scanner può ruotare continuamente, rendendo così l'acquisizione totale molto veloce (intorno ad 1 secondo).
Grazie al supporto tecnologico, nella quarta generazione migliaia di rilevatori circondano il paziente completamente così solamente la sorgente dei raggi X deve ruotare e ora il processo totale richiede meno di 1 secondo (figura 2.1(d)).
Negli anni 2000 è stato introdotto un nuovo approccio basato su un fascio di elettroni, questo tipo di CT è stato sviluppato per le immagini cardiache che sono caratterizzate da movimento a causa del battito cardiaco. A partire dalla terza e quarta generazione sono stati progettati vari CT scanner con differenti configurazioni e geometrie in base all'area specifica del corpo umano o per ricostruire sezioni umane più ampie.

2.1 Trasformata di Radon e la sua inversa

Nella tecnica di imaging dei raggi X, i dati di proiezione riflettono l'assorbimento dei fotoni di cui sono costituiti i raggi X, mentre la visualizzazione dell'oggetto corrisponde a un'immagine della mappa dei coefficienti di attenuazione

2.2 Legge di Lambert Beer

Tutti i meccanismi fisici, che portano all'attenuazione dell'intensità della radiazione misurata da un rilevatore dietro ad un oggetto omogeneo, sono solitamente ricondotti ad un singolo coefficiente di attenuazione $\mu = \mu(w) >= 0$ in base al punto attraversato w. All'interno di questo semplice modello è possibile calcolare l'attenuazione totale di un fascio di raggi X nel seguente modo:
l'intensità di radiazione misurata dopo aver passato uno spessore Δw attraverso un oggetto è determinata da

$$m(w + \Delta w) = m(w) - \mu(m(w)\Delta w) \tag{2.1}$$

dove $m(w)$ è l'intensità del fascio in arrivo. Riordinando l'equazione otteniamo:

$$\frac{m(w+\Delta w)-m(w)}{\Delta w} = -\mu(w)m(w) \tag{21.2}$$

Aggiungendo i limiti otteniamo il quoziente differenziale:

$$\lim_{\Delta w \to 0} \frac{m(w+\Delta w)-m(w)}{\Delta w} = \frac{dm}{dw} = -\mu(w)m(w) \tag{2.3}$$

Assumendo che l'oggetto è omogeneo si può descrivere con una singola costante di attenuazione, $\mu(w) = \mu$ lungo l'intero percorso Δw. Questo ci permette di trattare con una equazione differenziale di primo ordine e omogenea con coefficiente costante, quindi la sua soluzione può essere derivata per separazione di variabili. Nell'ultima equazione possiamo perciò separare:

$$\frac{dm}{m(w)} = -\mu dw \tag{2.4}$$

Con l'integrazione da entrambi i lati otteniamo:

$$\int \frac{dm}{m(w)} = -\mu \int dw \tag{2.5}$$

Fornendo:

$$\ln|m| = -\mu w + C \tag{2.6}$$

A causa della loro proprietà fisica, tutte le intensità m misurate sono quantità positive, quindi il valore assoluto può essere rimosso e si può applicare l'esponenziazione:

$$m(w) = e^{-\mu w + C} \tag{2.7}$$

Dato che la condizione iniziale $m(0) = m_0$ è il conteggio dei fotoni emessi, è noto per ogni fascio del macchinario CT e la soluzione della equazione differenziale (1.3) diventa:

$$m(w) = m_0 e^{-\mu w} \tag{2.8}$$

che è noto come legge di attenuazione di Lambert Beer.
In realtà, poiché il coefficiente di attenuazione lineare è una combinazione additiva di un coefficiente di dispersione e di un valore di assorbimento, la legge di Beer vale solo per il fascio a matita, perché qui la radiazione è diffusa completamente rimossa dal fascio principale. Tuttavia rimane ancora un concetto di base per la CT a raggi X.
Da un punto di vista matematico, la complicata dipendenza del coefficiente di attenuazione e del materiale penetrato fa sì che l'equazione differenziale (1.3) non possa essere completamente integrata come in (1.5).
In caso di attenuazione spazialmente variabile $\mu(w)$ la soluzione per l'intensità misurata dopo una lunghezza percorsa W è data da:

$$m(W) = m_0 e^{-\int_0^W \mu(w)dw} \tag{2.9}$$

Da qui si può facilmente derivare l'integrale della proiezione:

$$P(W) = -\ln(\tfrac{m(W)}{m_0}) = \int_0^W \mu(w)dw \tag{2.10}$$

Che è essenzialmente il logaritmo negativo del rapporto tra il numero di fotoni in uscita e in entrata per l'oggetto scansionato.

Questa legge funziona solo per fasci monoenergetici, che è un concetto abbastanza teorico, a causa della dipendenza energetica dei valori di attenuazione nella lunghezza d'onda dei raggi X emessi, dovremmo integrare tutte le energie del fascio:

$$m(W) = \int_0^{E_{max}} m_0(E) e^{-\int_0^W \mu(E,w)dw} dE \tag{2.11}$$

e considerando un fascio a raggi X policromatico. L'assunzione di raggi monocromatici e l'equazione (1.9) invece della (1.11) è l'origine di quello che viene chiamato *beam-hardening artifact.*

2.3 Radon

Nel 1917, il matematico austriaco Johann Radon fornì la base matematica per la ricostruzione di immagini tomografiche.
Impostiamo una geometria a matita, nel quale il fascio di raggi X viene collimato a forma di matita e mosso linearmente in parallelo a un array lineare di rilevatori di raggi X, per ogni direzione di proiezione, come era nella prima generazione di CT scanner. Fissiamo il caso bidimensionale, dove viene esaminata una singola fetta dell'oggetto alla volta, in questo caso possiamo identificare ogni punto w con una coppia bidimensionale (x; y) di coordinate e il coefficiente di attenuazione è una funzione continua e reale $\mu(w) = \mu(x, y)$ sul dominio spaziale della fetta, mappando di cosa è fatta. Identifichiamo un raggio che attraversa il percorso L e l'angolo d'emissione ϕ, come mostrato nella figura 2.2

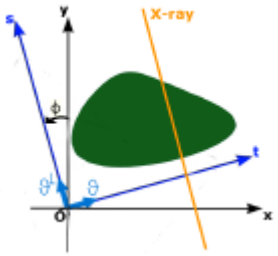


Figura 2.2: Rappresentazione di un processo CT su piano cartesiano.

Adesso, l'integrale di proiezione nell'equazione (1.10) rappresenta una integrazione lungo il percorso L, attraversato dai fotoni emessi, ed è definito dalla posizione del sorgente dei raggi X e dei rilevatori. Il valore registrato m è quindi:

$$-\ln(\tfrac{m}{m_0}) = +\int_L \mu(x, y)dw \tag{2.12}$$

Consideriamo il fascio parallelo nel sistema cartesiano tOs con versore $\theta = (cos\phi, sin\phi)$ e $\theta^{\perp} = (-sin\phi, cos\phi)$, come mostrato nella figura 2.2. Per ogni raggio X, L è una funzione di ϕ(o θ) e t, quindi possiamo riformulare l'integrale in (1.12) nelle nuove coordinate, ottenendo:

$$\int_L \mu(x, y)dw = \int_{-\infty}^{+\infty} \mu(t\theta + s\theta^{\perp})ds = P_0(t) \tag{2.13}$$

Che rispecchia perfettamente la proiezione P dell'oggetto lungo un raggio θ inclinato in t. D'ora in poi, θ rappresenta anche l'angolo di scansione, perché determina ϕ unicamente.

Dato l'angolo di scansione θ, la trasformata di Radon di μ è definita come la mappa $R_\theta : \mu(x, y) -> P_\theta$, tale che

$$(R_\theta \mu)(t) = \int_{-\infty}^{+\infty} \mu(t\theta + s\theta^{\perp}) ds \qquad \forall \text{ t } \in \Re \tag{2.14}$$

Significa che la trasformata di Radon R_θ di un oggetto, descritta da μ, è la proiezione completa P_θ dell'oggetto stesso, quando è scansionata interamente dall'angolo θ.
Il processo circolare della CT si basa su una continua acquisizione, il che significa che una trasformata di Radon della fetta di interesse viene misurata sul rilevatore da tutti gli angoli $\theta \in [-\pi, \pi]$. I dispositivi possono compiere solo piccoli passi angolari $\Delta k \in \theta_1, \ldots, \theta_{n\theta}$ e a causa della nitidezza delle componenti del rilevatore anche le proiezione sono registrate in un numero finito di punti t_i dove i=1, ..., N_p.
La rappresentazione grafica di tutti i dati è chiamata sinogramma

2.4 Backprojection operator

Una volta raccolti tutti i dati di proiezione, si definisce il problema matematico inverso: come tornare alla distribuzione spaziale dei coefficienti di attenuazione $\mu(w)$, avendo misuro un certo numero N_θ di dati di proiezione P_θ?
L'idea base è di proiettare all'indietro ogni dato fino al suo percorso di raggio originale, come mostrato nella figura 2.3.

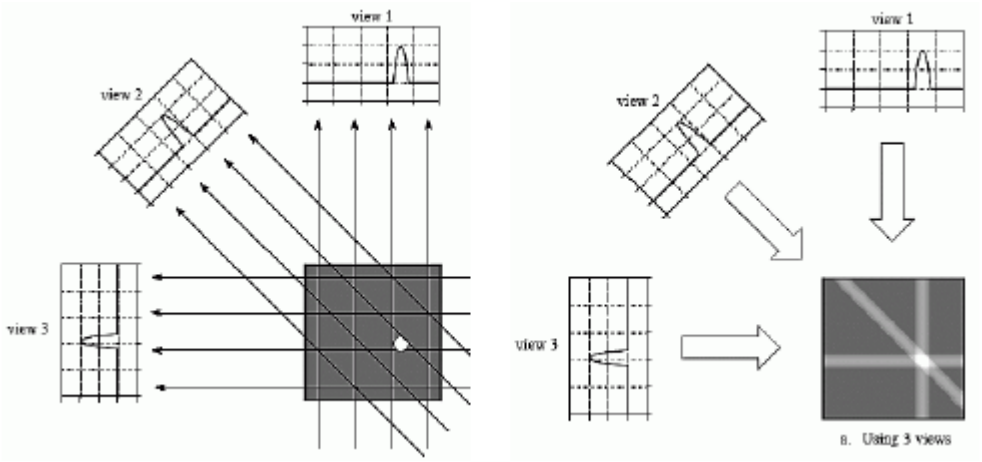


Figura 2.3: Tre proiezione sono acquisite per una singola fetta. Successivamente essi sono back-projected dentro un'immagine vuota, per mostrare come una Back-Projection funziona, con $N_\theta = 3$.

In questo punto sorgono molti problema per l'implementazione pratica. Prima di tutto abbiamo bisogno di sapere esattamente quali punti sono coinvolti per ogni dato e tracciare tutti i raggi X è una task costosa. Secondo, i dati reali sono danneggiati dal rumore, la cui propagazione deve essere affrontata durante la fase di ricostruzione.
I primi software commerciali erano basati su un approccio analitico: tenendo conto del teorema di Fourier Slice è possibile saltare la fase pesante del ray-tracing, per conto della trasformata di Fourier per ogni trasformata di Radon. Inoltre, nel dominio della frequenza è possibile applicare opportuni filtri di smoothing e ridurre le alte frequenze che tipicamente enfatizzano la propagazione del rumore. Queste caratteristiche definiscono il noto algoritmo di Filtered Back Projection (FBP) che è stato ampiamente utilizzato e sviluppato in molti software commerciali negli ultimi decenni.

Capitolo 3

Rete neurale convoluzionale encoder-decoder residua per ct a basso dosaggio

3.1 Modello di riduzione del rumore

Il nostro flusso di lavoro inizia con una semplice ricostruzione FBP da una scansione a bassa dose (LDCT) e il problema di riduzione del rumore dell'immagine è limitato all'interno del dominio dell'immagine. Poiché i metodi basati su DL sono indipendenti dalla distribuzione statistica del rumore, il problema LDCT può essere semplificato come segue. Assumendo che $\mathrm{X} \in R^{mxn}$ sia un'immagine LDCT e $\mathrm{Y} \in R^{mxn}$ è la corrispondente immagine a dose normale (NDCT), la loro relazione può essere formulata come:

$$X = \sigma(Y) \tag{3.1}$$

dove $\sigma : R^{mxn} \to R^{mxn}$ denota il complesso processo di degradazione che coinvolge il rumore e altri fattori. Quindi, il problema può essere trasformato per cercare una funzione f:

$$\underset{f}{\operatorname{argmin}} \, ||f(X) - Y||_2^2 \tag{3.2}$$

dove f è considerata l'approssimazione ottimale di σ^{-1} e può essere stimata utilizzando tecniche DL.

3.2 Rete autoencoder residua

L'autoencoder (AE) è stato originariamente sviluppato per l'apprendimento senza supervisione di funzioni da input con rumore, che è anche adatto per il ripristino delle immagini. Nel contesto della riduzione del rumore anche la CNN ha dimostrato un'ottima performance. Tuttavia, a causa delle sue molteplici operazioni di sottocampionamento alcuni dettagli dell'immagine posso essere ignorati dalla CNN. Per le LCDT, verrà proposta una rete residua che unisce AE e CNN. Piuttosto che adottare livelli FC per la codifica e decodifica, utilizziamo livelli sia convoluzionali che deconvoluzionali in simmetria. Inoltre, a differenza della tipica struttura encoder-decoder, è previsto l'apprendimento residuo con scorciatoie per facilitare le operazioni dei livelli convoluzionali e dei corrispondenti livelli deconvoluzionali.
L'architettura complessiva della rete residua encoder-decoder (RED-CNN) è mostrata nella figura 3.1. Questa rete contiene 10 livelli, di cui 5 convoluzionali e 5 deconvoluzionali disposti simmetricamente. Le scorciatoie collegano i livelli convoluzionali e deconvoluzionali corrispondenti.

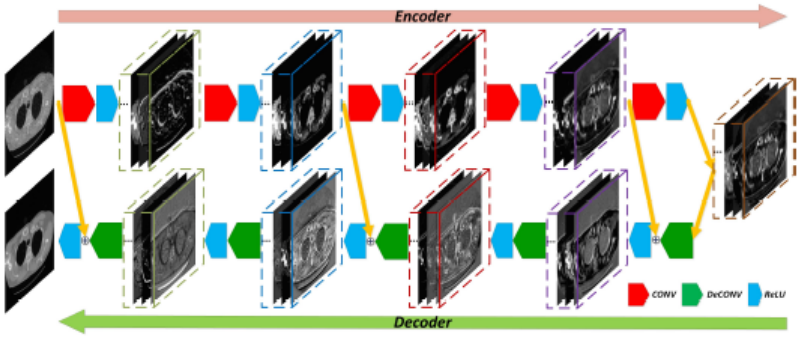


Figura 3.1: L'immagine rappresenta 5 encoder in pila e 5 decoder in pila in simmetria di una rete neurale convoluzionale che sfrutta una rete autoencoder residua.

La rete è divisa in 5 fasi:

1. **Estrazione delle Patch:** I metodi basati su DL richiedono un numero enorme di campioni. Questo requisito non può essere soddisfatto facilmente nella pratica, specialmente per l'imaging medico. Qui, proponiamo di utilizzare patch sovrapposte nelle immagini CT. Questa strategia si è rilevata efficace ed efficiente, poiché è possibile rilevare le

differenze percettive delle regioni locali e il numero di campioni è notevolmente aumentato. Negli esperimenti verranno estratte patch da LDCT e le corrispondenti immagini NDCT con una dimensione fissa.

2. **Encoder in pila (riduzione del rumore e degli artefatti):** A differenza delle tradizionali reti AE in pila, verrà usata una catena di livelli convoluzionali FC come codificatori in pila. I rumori e gli artefatti vengono eliminati passo dopo passo da quelli di basso livello fino a quelli di alto livello per preservare le informazioni essenziali nelle patch estratte. Inoltre, poiché il livello Pooling dopo un livello convoluzionale può scartare importanti dettagli strutturali, non è presente nel codificatore. Di conseguenza nel nostro codificatore ci sono solo due strati, quello convoluzionale e quello ReLU, e il codificatore in pila $C_e^i(x_i)$ può essere formulato come:

$$C_e^i(x_i) = ReLU(W_i * x_i + b_i) \qquad \text{i = 0, 1, ..., N} \tag{3.3}$$

Dove N è il numero dei livelli convoluzionali, W_i e b_i indicano rispettivamente i pesi e le distorsioni, $*$ rappresenta l'operatore di convoluzione, x_0 è la patch estratta dalle immagini di input e x_i (i > 0) è le caratteristiche estratte dai livelli precedenti. $ReLu(x) = max(0, x)$ è la funzione di attivazione. Dopo i codificatori in pila, le patch dell'immagine sono trasformate in uno spazio delle caratteristiche e l'output è un vettore delle caratteristiche x_N la cui dimensione è l_N.

3. **Decoder in pila (ripristino dei dettagli strutturali):** Sebbene l'operazione di Pooling è rimossa, una serie di convoluzioni, che essenzialmente agiscono come filtri di rumore, ridurranno comunque i dettagli dei segnali in input.
In base ai risultati sulla segmentazione semantica e sulla segmentazione di immagini biomediche, i livelli deconvoluzionali sono integrati nel nostro modello per il recupero dei dettagli strutturali, che può essere visto come la ricostruzione dell'immagine da caratteristiche estratte, questi livelli deconvoluzionali sono FC.
Poiché i codificatori e i decodificatori dovrebbero apparire in coppia, i livelli convoluzionali e deconvoluzionali sono simmetrici nella rete proposta. Per garantire che l'input e l'output della rete corrispondano esattamente, i livelli convoluzionali e deconvoluzionali devono avere la stessa dimensione del kernel. In questa rete il flusso di dati attraverso i livelli convoluzionali e deconvoluzionali segue la regola del "FILO" (First In Last Out). Come dimostrato nella figura 3.1, il primo livello di convoluzione corrisponde all'ultimo livello di deconvoluzione, l'ultimo

livello di convoluzione corrisponde al primo livello di deconvoluzione e così via. In altre parole, questa architettura è caratterizzata dalla simmetria dei livelli convoluzionali e deconvoluzionali accoppiati. Esistono due livelli nella rete di decodificato, quello di deconvoluzione e quello ReLU. Pertanto i decodificato in pila $D_d^i(y_i)$ possono essere formulati come:

$$D_d^i(y_i) = ReLU(W_i^{'} \otimes y_i + b_i^{'}) \tag{3.4}$$

dove N è il numero di strati deconvoluzionali, W_i e b_i indicano i pesi e le distorsioni, $\otimes$ rappresenta l'operatore deconvoluzionale, $y_N = x$ è il vettore della caratteristica di output dopo la codifica in pila, y_i (N > i >0) è il vettore della caratteristica ricostruita dal precedente strato deconvoluzionale, e y_0 è la patch ricostruita. Dopo la decodifica in pila, le patch dell'immagine vengono ricostruite dalle caratteristiche e possono essere assemblate per ricostruire un'immagine senza rumore.

4. **Compensazione residua:** La convoluzione eliminerà alcuni dettagli dell'immagine e nonostante la deconvoluzione possa recuperare alcuni dettagli, quando la rete va più in profondità la perdita accumulata potrebbe essere abbastanza insoddisfacente per la ricostruzione dell'immagine. Inoltre, quando la profondità della rete aumenta, la diffusione del gradiente potrebbe rendere la rete difficile da addestrare.
   Per affrontare questo problema viene utilizzato un meccanismo di compensazione residua. Invece di mappare l'input all'output esclusivamente dagli strati sovrapposti, viene utilizzata una mappatura residua, come mostrato nella figura 3.2.

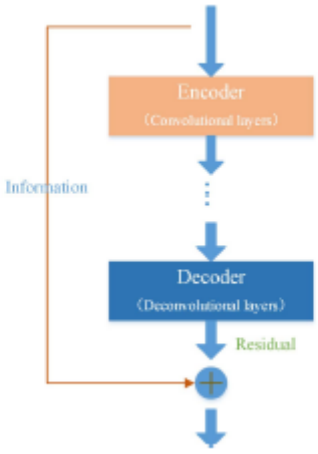


Figura 3.2: L'immagine rappresenta il processo che da una immagine in input si passa dagli encoder e dai decoder ottenendo alla fine la mappa residua.

Definendo l'input come I e l'output come O, la mappatura residua può essere indicata come $F(I) = O \ominus I$, e usiamo livelli sovrapposti per adattarsi a questa mappatura. Una volta che la mappatura residua è stata costruita, possiamo ricostruire la mappatura originale come $R(I) = O = F(I) + I$. Di conseguenza, trasformiamo il problema di mappatura diretta in un problema di mappatura residua.
Ci sono due vantaggi associati alla mappatura dei residui. In primo luogo, è più facile ottimizzare la mappatura residue che ottimizzare la mappatura diretta. Ciò aiuta ad evitare che il gradiente svanisca quando la rete è profonda. Ad esempio, sarebbe molto più facile addestrare una rete di mappatura dell'identità spingendo il residuo a zero che adattare direttamente una mappatura dell'identità. In secondo luogo, poiché solo il residuo viene elaborato dai livelli convoluzionali e deconvoluzionali, è possibile preservare più dettagli strutturali e di contrasto negli output dei livelli deconvoluzionali, il che può migliorare significativamente le prestazioni di imaging LDCT.
Le scorciatoie sono utilizzate sia per la conservazione dei dettagli strutturali che per facilitare la formazione di reti più profonde. Inoltre, la struttura simmetrica delle coppie dei livelli di convoluzione-deconvoluzione permette di mantenere più dettagli mentre si sopprimono il rumore e gli artefatti.
Per la segmentazione viene utilizzato sia le scorciatoie che la deconvoluzione. Le caratteristiche ad alta risoluzione sono combinate con un output sovracampionato per migliorare la classificazione dell'immagine.

5. **Training:** La rete proposta è una mappatura end-to-end da CT a bassa dose a CT a dose normale. Una volta configurata la rete, l'insieme dei parametri, $\Theta = \{W_i, b_i, W_i^{'}, b_i^{'}\}$, dei livelli convoluzionali e deconvoluzionali dovrebbero essere stimati per costruire la funzione di mappatura M. La stima può essere ottenuta minimizzando la perdita $F(D;\Theta)$ tra le immagini CT stimate e la corrispondente immagine NDCT X. Dato un insieme di patch accoppiate $P = (X_1, Y_1), (X_2, Y_2), \ldots, (X_k, Y_k)$ dove X_i e Y_i denotano rispettivamente le patch delle immagini NDCT e LDCT e K è il numero totale dei campioni di addestramento. L'errore quadratico medio (MSE) viene utilizzato come funzione di perdita:

$$F(D;\Theta) = \frac{1}{N}\sum_{i=1}^{N} ||X_i - M(Y_i)||^2 \tag{3.5}$$

3.3 Rete U-Net

Un altro tipo di rete che sarà utilizzata per i test è la rete U-Net. La rete U-Net è stata sviluppata per applicazioni in campo medico, come l'individuazione di tumori nei polmoni e nel cervello. L'U-Net è costituita da un encoder e un decoder. L'encoder riduce l'immagine in ingresso in una feature map estraendone gli elementi chiave. Il decoder amplifica la feature map in una immagine. Il nome U-Net deriva dalla forma della sua struttura che si può vedere nella figura 3.3.

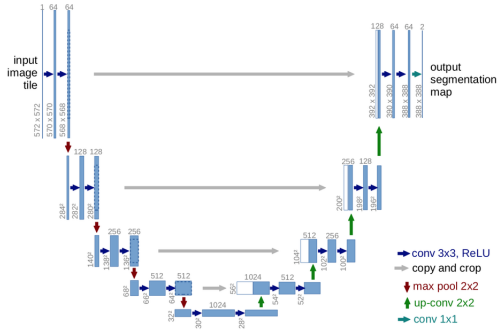


Figura 3.3: Ogni blu box corrisponde ad una feature map multicanale. Il numero di canali è specificato al di sopra dei box. Le dimensioni x-y sono indicate nell'angolo in basso a sinistra di ogni box. Quelli bianchi sono invece le feature map copiate. Le frecce indicano le differenti operazioni eseguite.

Questa struttura la rende particolarmente adatta a risolvere problemi di segmentazione delle immagini. L'encoder, o percorso di contrazione, cattura il contesto dell'immagini, esso è costituito dai livelli di convoluzione e di Max Pooling. Il decoder, o percorso di espansione, localizza con precisione gli elementi dell'immagine attraverso le convoluzioni trasposte. Ogni espansione del decoder riceve i dati dalla corrispondente contrazione dell'encoder, poichè i livelli iniziali dell'encoder contengono più informazioni garantiscono un significativo miglioramento nel processo di espansione permettendo il recupero di dettagli e migliorando significativamente il risultato. Queste connessioni sono chiamate shortcut, in particolare la prima shortcut crea un ponte tra l'encoder prima dell'iniziale filtro di pooling e il decoder dopo l'ultima operazione di deconvoluzione.

3.4 Specifiche rete

Per le prove vengono utilizzate due tipi di rete, una rete U-Net e una rete RED-CNN.
La rete U-Net è composta da 4 layer sia per la parte encoder e sia per la parte decoder. I primi due livelli hanno entrambi dimensione 32, mentre il terzo 64 e il quarto 128. Tutti i livelli hanno il passo uguale ad 1 ed è presente un Max Pooling in tutti i livelli tranne che nel primo. Nel lato decoder le dimensioni dei livelli sono simmetriche rispetto a quelle dell'encoder ed è presente una operazione di *concatenate* come opposto al Max Pooling, però qui il passo è sempre 2 tranne nell'ultimo livello che è uguale ad 1. Per tutti i livelli dell'encoder e per l'ultimo del decoder la funzione di convoluzione ha come parametro di attivazione *tanh*.
Invece la rete RED-CNN è composta da 5 livelli per la fase encoder e 5 livelli per la fase decoder. Nella parte encoder la dimensione è 96 per tutti i livelli e il passo è sempre 1. In questa rete, rispetto alla rete U-Net, la parte decoder è simmetrica alla parte encoder, oltre che per le dimensioni dei vari livelli anche per il passo.

Capitolo 4

Risultati numerici

4.1 Training

Nei vari test sono stati utilizzati tre tipi di training che chiameremo T1, T2 e T3.
Il T1 corrisponde ad un dataset di 1000 immagini sintetiche, queste immagini comprendono ellissi che rappresentano masse con diverse contrasto in una tomografia, linee per le fibre e puntini per le microcalcificazioni.
Il T2 corrisponde ad un dataset di 300 immagini reali prese dal sito National Biomedical Imaging Archive (NBIA) e che rappresentano l'Head-Neck Cetuximab.
Il T3 corrisponde ad un dataset misto, composto da immagini sintetiche e da immagini reali. Per questo training il numero di immagini nel dataset cambieranno in base ai risultati ottenuti.
Tutte le immagini avranno una dimensione di 256 x 256 pixel.
Le immagini originali saranno passate alla rete che provvederà a corrompere le immagini tramite l'applicazione della trasformata di radon su 100 angoli. Successivamente verrà aggiunto del rumore gaussiano con il seguente comando:

*sinograms = sinograms + 0.5*np.random.randn(sinograms.shape[0], sinograms.shape[1])*

Dove sinograms è il risultato della precedente trasformata di radon.
Infine applichiamo l'inversa di radon per ottenere la FBP dell'immagine e salviamo il dataset con le immagini corrotte.

Una volta ottenuto il dataset con le immagini originali e quello con le immagini corrotte viene effettuato il training delle due reti. Il training sarà effettuato con un batch size di 20 e su 200 epoche. Quando la rete è addestrata verranno passato delle immagini per testare il training, queste immagini

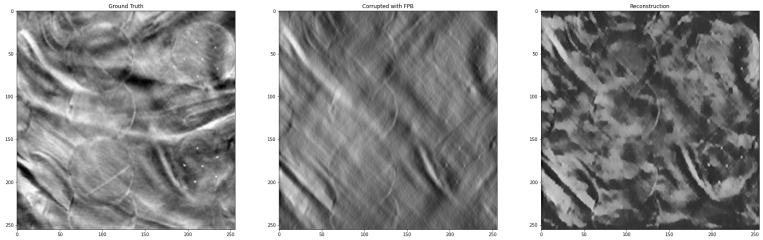
saranno corrotte su 50 o 100 angoli, in base al test che si vuole effettuare, e poi ricostruite dalla rete in esame.

4.2 Test su immagini sintetiche

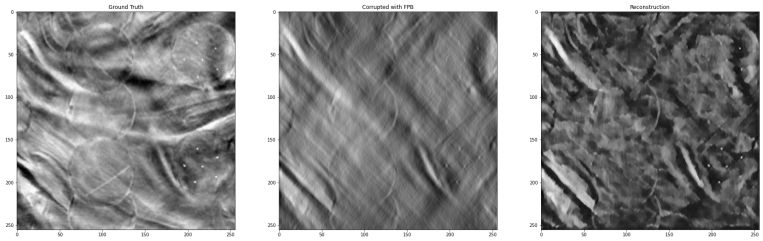
I primi test effettuati riguardano il training T1.
Le immagini utilizzate per testare il training sono 3. La prima è una simulazione di una tomografia al seno, nella quale sono presenti delle sfere acriliche, delle fibre e delle microcalcificazioni. La seconda è una immagine sintetiche composta da masse di diverse contrasto, mentre la terza è una immagine del dataset originale.
I risultati di ogni test saranno composti da 3 immagini, a sinistra l'immagine originale, nel mezzo l'immagine corrotta e a destra l'immagine ricostruita dalla rete. Per ogni training verranno effettuati test sia con la rete RED-CNN che con la rete U-Net.

4.2.1 100 angoli

Nel primo training effettuato le immagini sono state corrotte su 100 angoli e i risultati sono i seguenti.

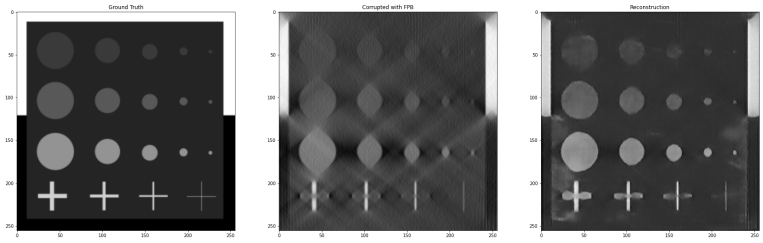


(a)

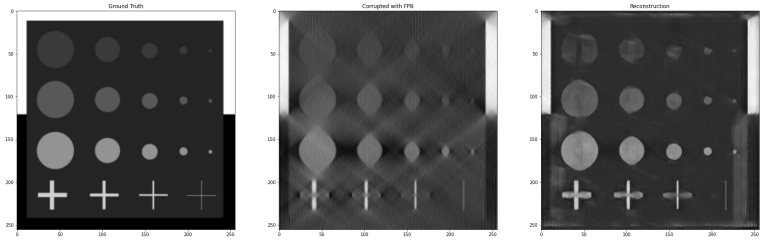


(b)

Figura 4.1: Training T1, immagini corrotte su 100 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. Si può notare che in entrambe le immagini si vedono le sfere acriliche e i puntini delle microcalcificazioni, l'unica differenza che può risaltare di più è che i puntini in alto sono più visibili nella rete RED-CNN.

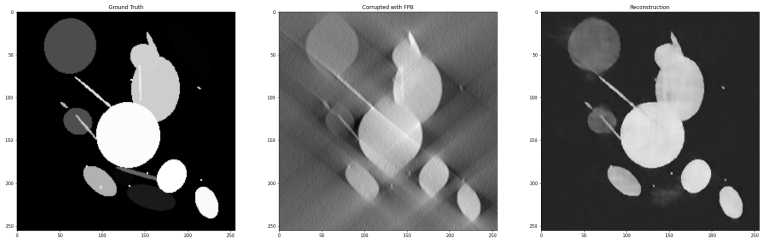


(a)

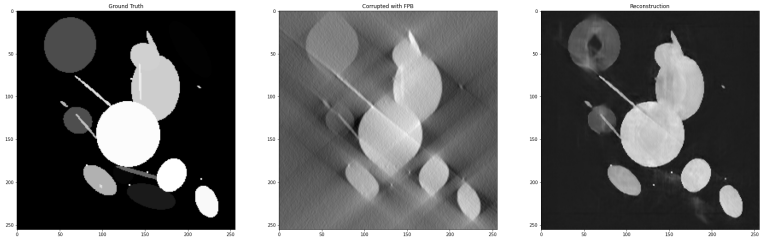


(b)

Figura 4.2: Training T1, immagini corrotte su 100 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. Entrambi le rete hanno riconosciuto tutte le masse, anche quelle con poco contrasto, però nella U-Net sono state ricostruite meglio, e nella RED-CNN quelle con poco contrasto sono un po' tagliate e la prima ha anche un vuoto nel mezzo.



(a)

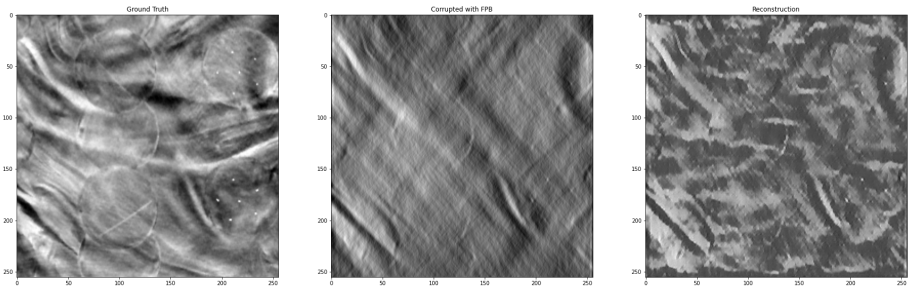


(b)

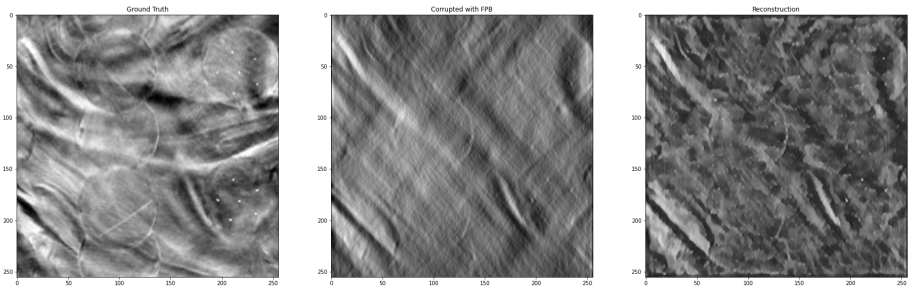
Figura 4.3: Training T1, immagini corrotte su 100 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. Entrambi le rete hanno rilevato le masse, le linee e i puntini, l'unica pecca è nella RED-CNN che c'è un buco nella ellissi in alto a sinistra, mentre nella U-Net è stato ricostruito tutto bene.

4.2.2 50 angoli

In questo training le immagini sono corrotte su 50 angoli.
Era stato svolto anche un training su 50 angoli per vedere se veniva ricostruito meglio rispetto al training su 100 angoli, ma dava risultati migliori il training su 100 angoli.

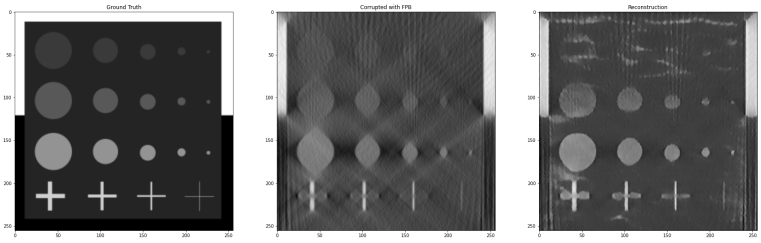


(a)

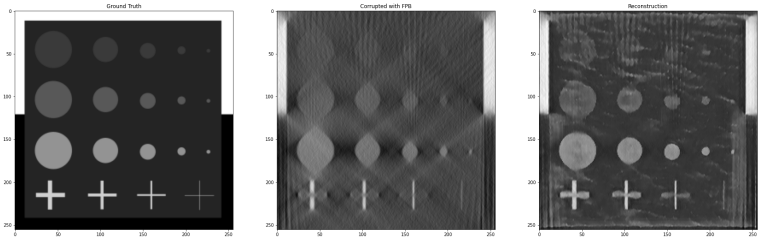


(b)

Figura 4.4: Training T1, immagini corrotte su 50 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. Rispetto al test effettuato con le immagini sporcate su 100 angoli, qui si sono persi alcuni dettagli, i puntini, in particolare quelli in alto, non sono stati presi tutti e le sfere non sono totalmente visualizzate.

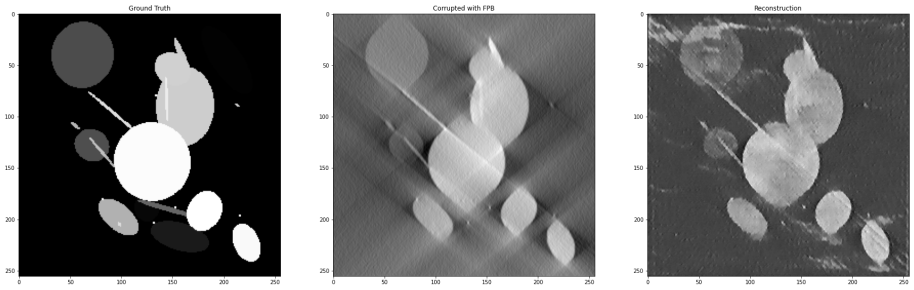


(a)

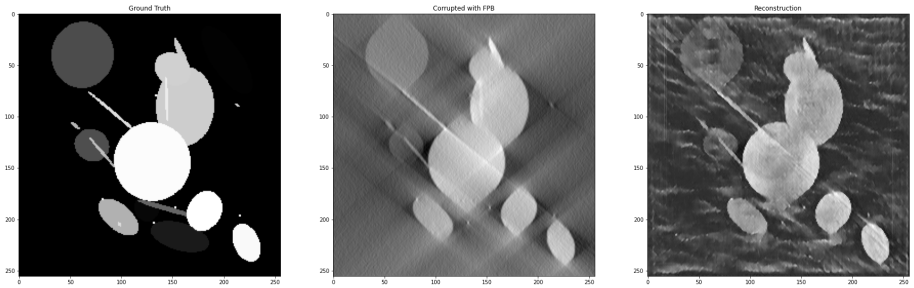


(b)

Figura 4.5: Training T1, immagini corrotte su 50 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. Anche per questa immagine si è ridotta la qualità della ricostruzione dell'immagine rispetto a quella su 100 angoli, in particolare le masse con poco contrasto sono difficilmente rilevate.



(a)



(b)

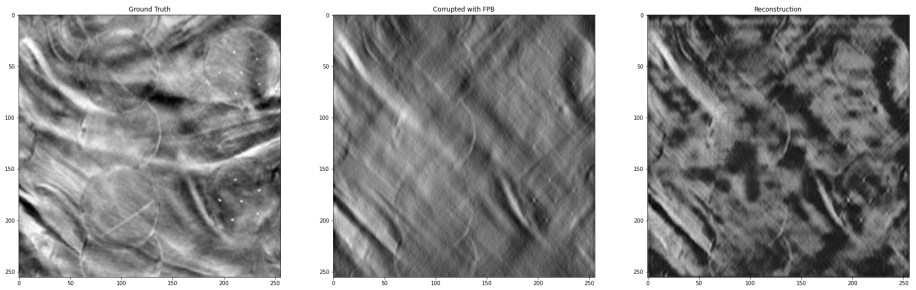
Figura 4.6: Training T1, immagini corrotte su 50 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. Anche in questo caso la qualità si è ridotta rispetto al test sui 100 angoli, e il risultato della U-Net è migliore rispetto a quello della RED-CNN.

4.3 Test su immagini reali

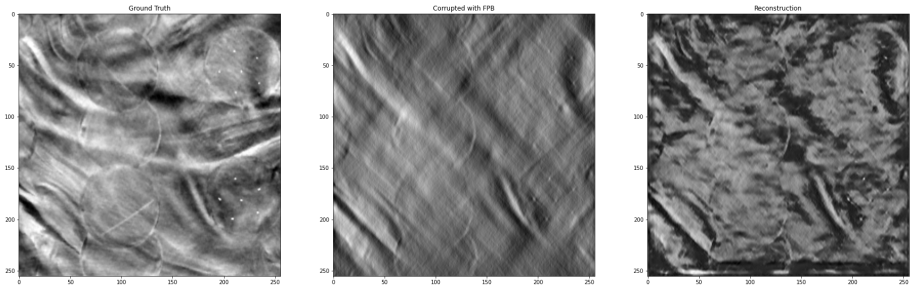
Questo nuovo test riguarda il training T2.
Anche in questo training le imamgini di testing sono 3 e sono sempre le stesse di prima. L'unica differenza è che l'immagine presa dal dataset sarà diversa, poichè il dataset è cambiato e non è più quello di immagini sintetiche.
Infatti il dataset sarà composto da 300 immagini riguardanti l'Head-Neck Cetuximab dal sito National Biomedical Imaging Archive (NBIA).

4.3.1 100 angoli

In questo test le immagini vengono corrotte su 100 angoli.

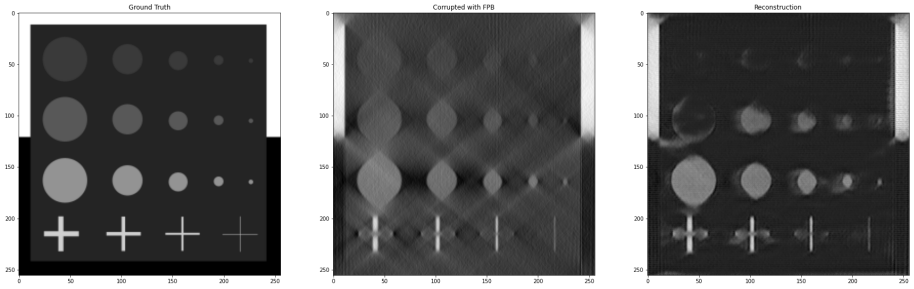


(a)

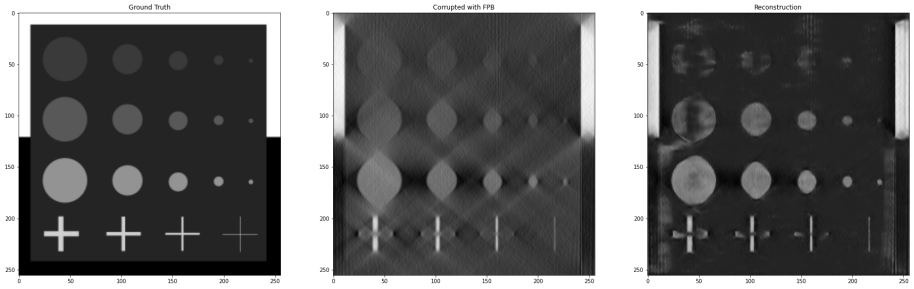


(b)

Figura 4.7: Training T2, immagini corrotte su 100 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. Rispetto al training con immagini sintetiche l'immagine è peggiorata, i cerchi ancora si vedono, ma in alcuni punti sono spariti, e stessa cosa anche per i puntini. La rete RED-CNN qui è meglio della U-Net.

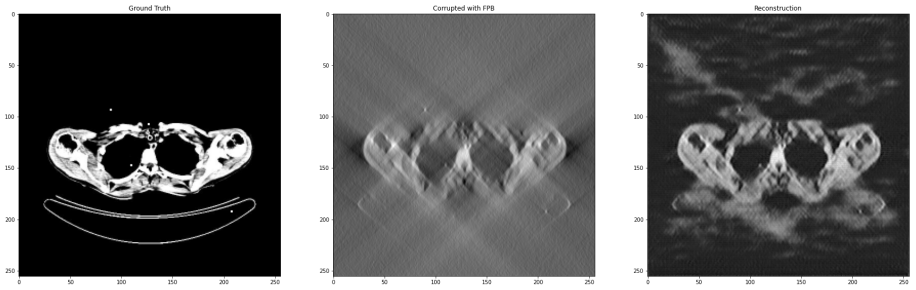


(a)

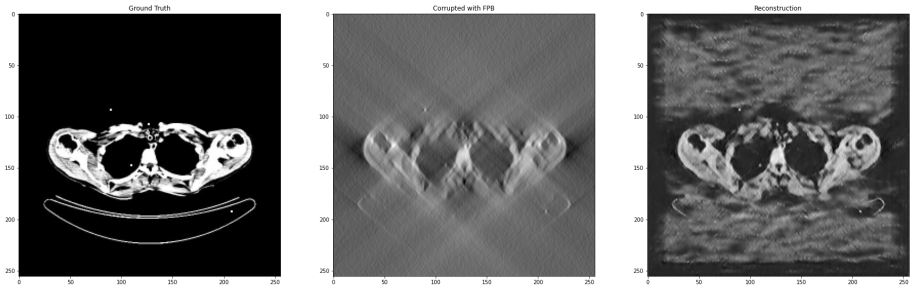


(b)

Figura 4.8: Training T2, immagini corrotte su 100 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. Rispetto al training con immagini sintetiche c'è stato un netto peggioramento, la prima riga è praticamente scomparsa e le masse più piccole sono appena visibili. La rete RED-CNN restituisce un risultato migliore rispetto a quella U-Net.



(a)

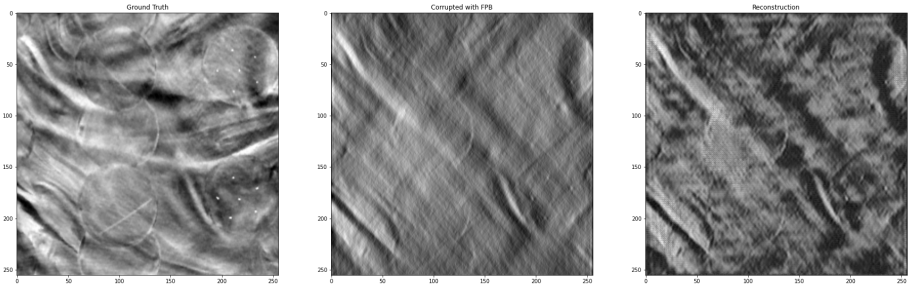


(b)

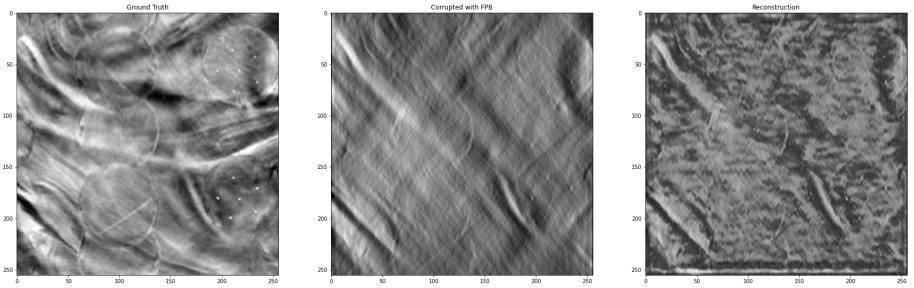
Figura 4.9: Training T2, immagini corrotte su 100 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. Le immagini sintetiche con questo training erano peggiorate, invece l'immagine reale è ben riprodotta, sebbene con qualche artefatto. Nella rete U-Net si può notare che c'è meno rumore rispetto alla RED-CNN.

4.3.2 50 angoli

Questa volta le immagini sono corrotte su 50 angoli.

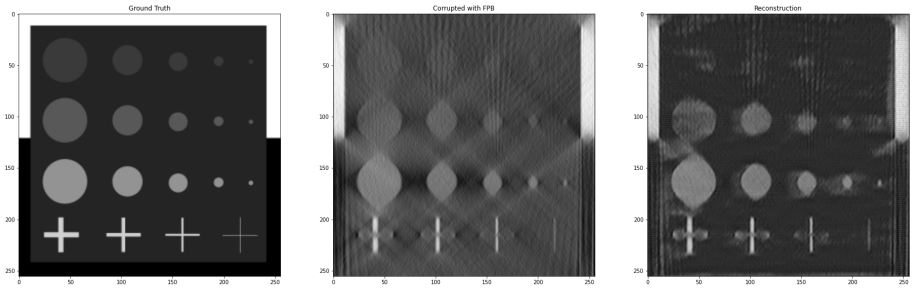


(a)

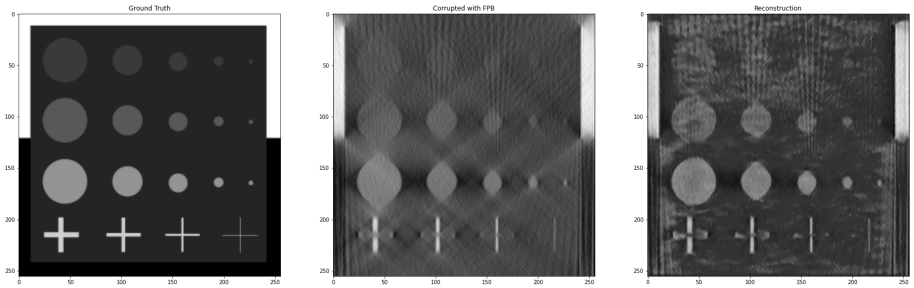


(b)

Figura 4.10: Training T2, immagini corrotte su 50 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. Le immagini rispetto a quelle corrotte su 100 angoli sono peggiorate e perdono molto dettagli. Nella rete RED-CNN si sono persi molto dettagli, anche di più rispetto alla U-Net.

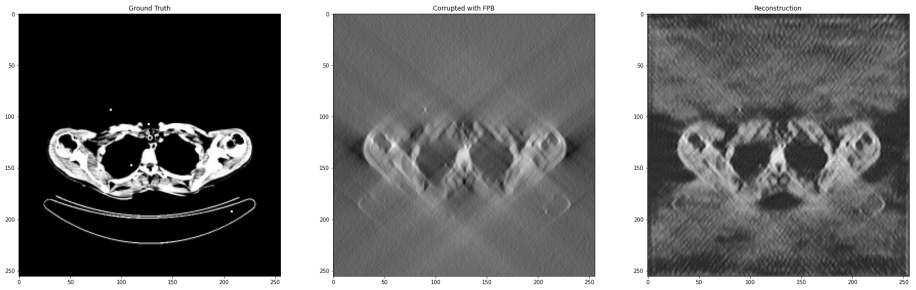


(a)

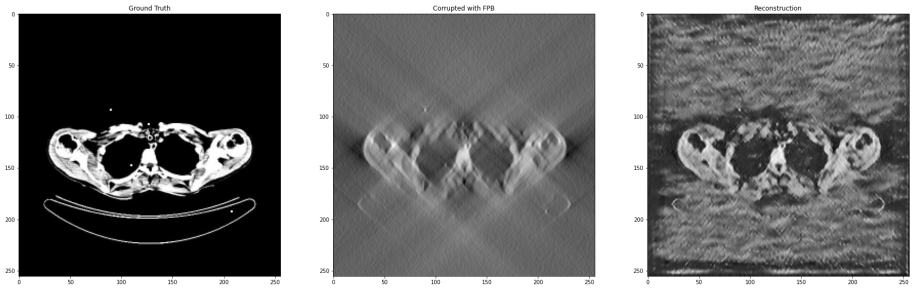


(b)

Figura 4.11: Training T2, immagini corrotte su 50 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. Anche in questo caso l'immagine è peggiorata, presenta molto rumore e nella rete U-Net la prima riga di masse è praticamente scomparsa.



(a)



(b)

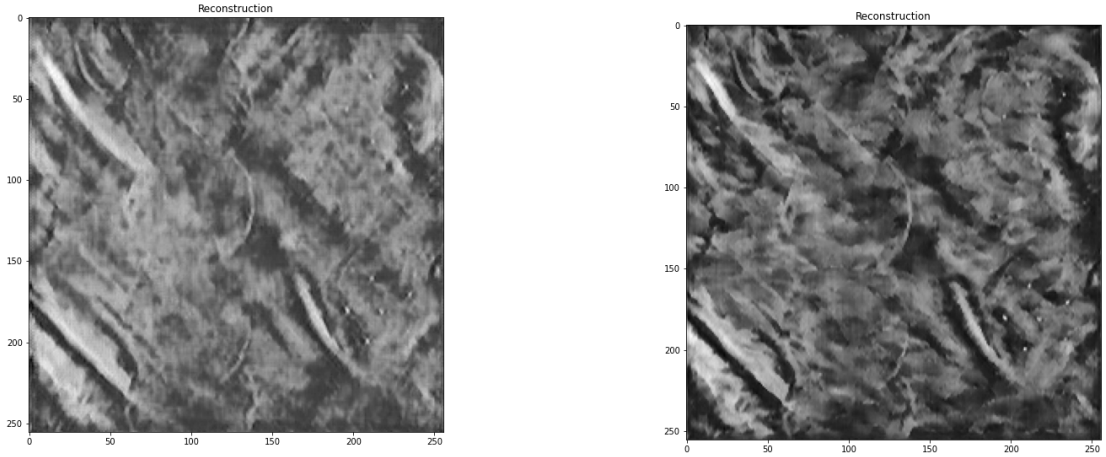
Figura 4.12: Training T2, immagini corrotte su 50 angoli, nella figura (a) rete U-Net, nella figura (b) rete RED-CNN. I puntini e l'oggetto dell'immagine sono ricostruiti, ma il resto dell'immagine presenta molto rumore. Nella rete U-Net è presente meno rumore rispetto alla rete RED-CNN.

4.4 Test su dataset misto

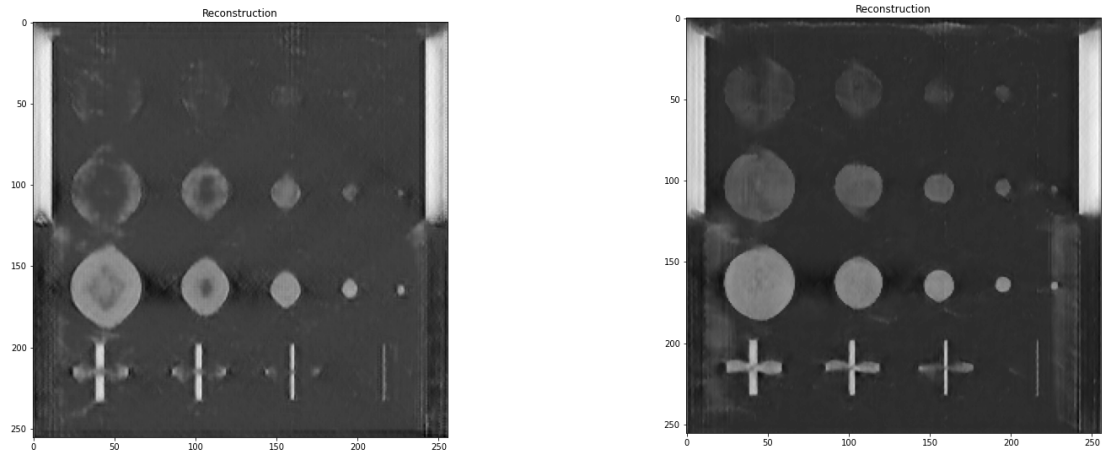
Questo nuovo test riguarda il training T3.
A seguito di questi risultati si è notato che il training con immagini sintetiche portava buoni risultati, quindi si è passato a provare un training con un dataset misto composto da immagini sintetiche e immagini reali.
In questi test saranno mostrate solamente le immagini ricostruite, a sinistra con la rete U-Net e a destra con la rete RED-CNN.

4.4.1 Primo test

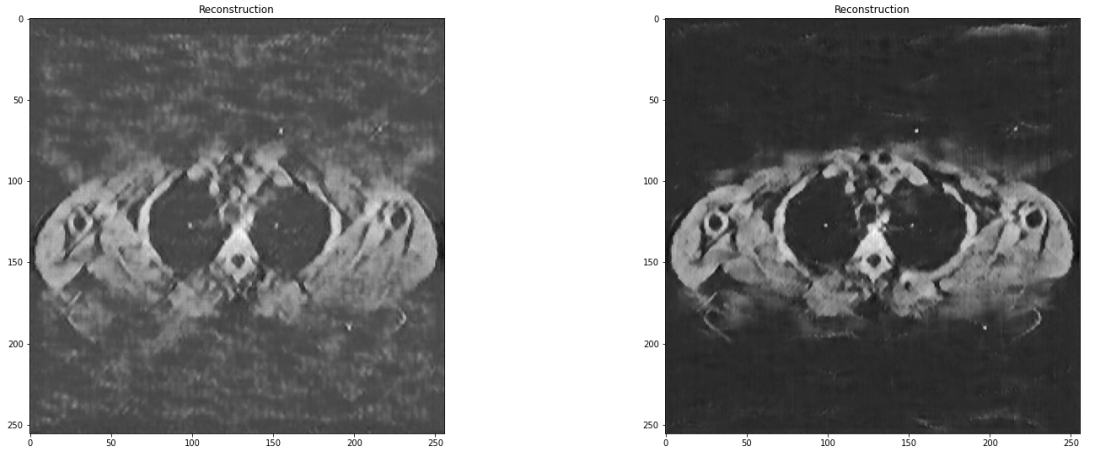
La prima prova è effettuata con 700 immagini sintetiche e 300 reali e restituisce i seguenti risultati.



(a)



(b)

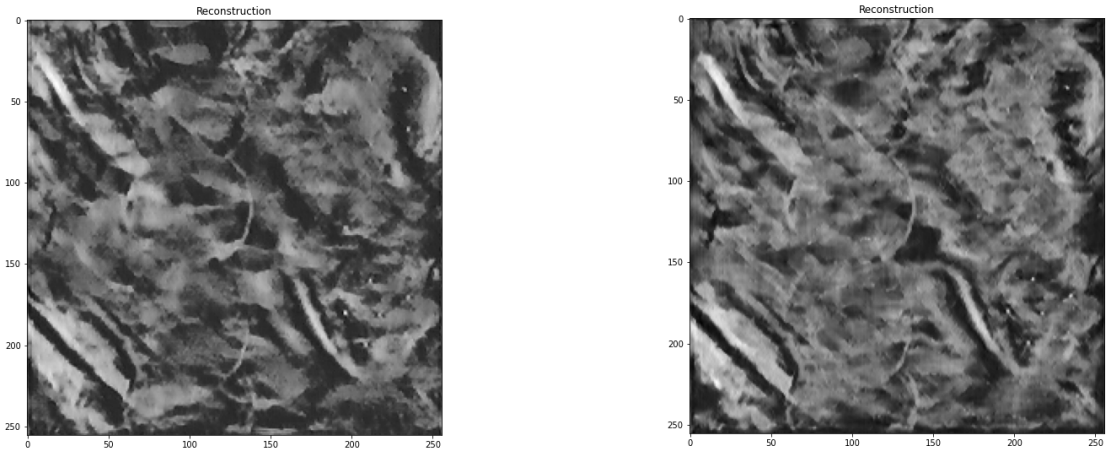


(c)

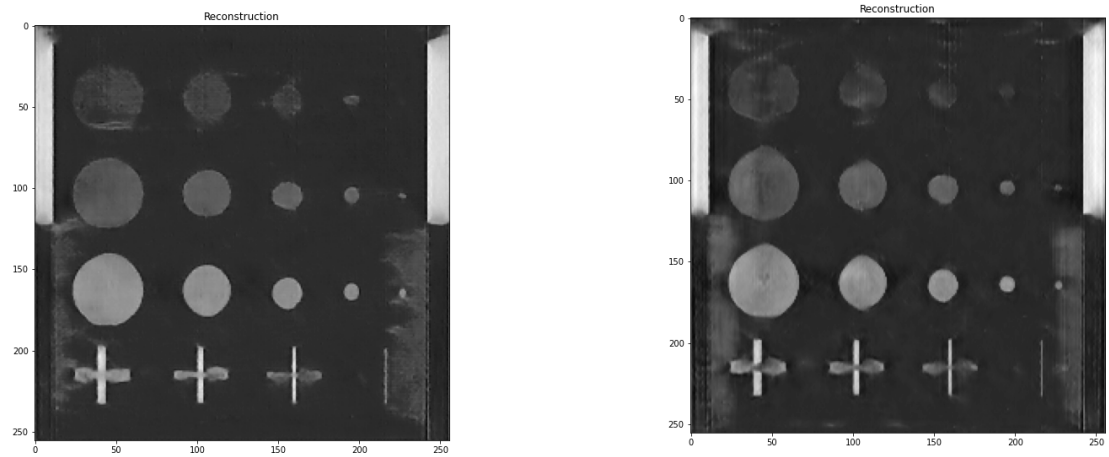
Figura 4.13: Training T3, 700 sintetiche, 300 reali, immagini corrotte su 100 angoli, a sinistra rete U-Net, a destra rete RED-CNN. In questo training rispetto al T2, le immagini sintetiche sono migliorate e l'immagine reale ha perso un po' di rumore. In questo caso la rete RED-CNN restituisce una immagine con meno rumore rispetto alla U-Net.

4.4.2 Secondo test

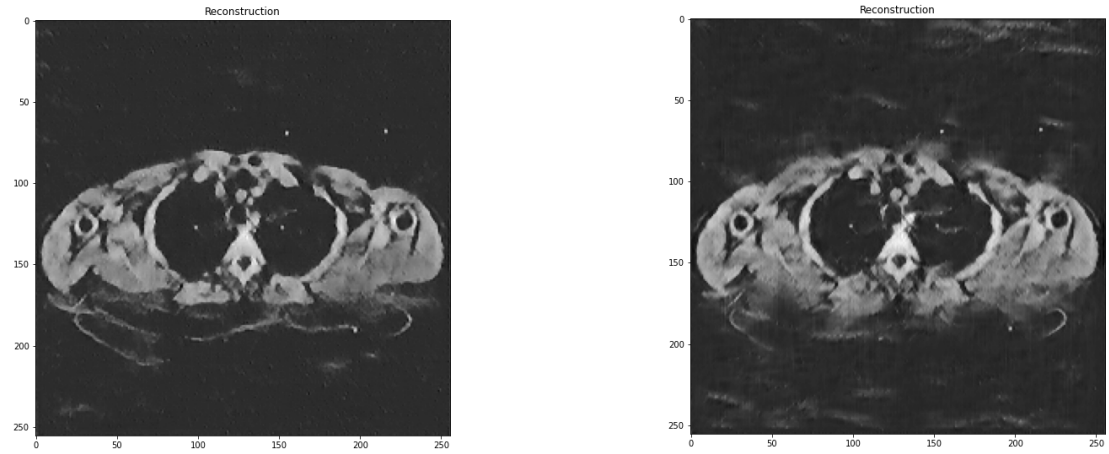
Si può notare che l'immagine è influenzata dalle immagini sintetiche e potrebbe essere resa più realistica. A seguito di queste considerazioni è stato effettuato un training con 500 immagini sintetiche e 300 immagini reali che ha restituito i seguenti risultati (a sinistra l'esito della rete U-Net, a destra delle rete RED-CNN).



(a)



(b)

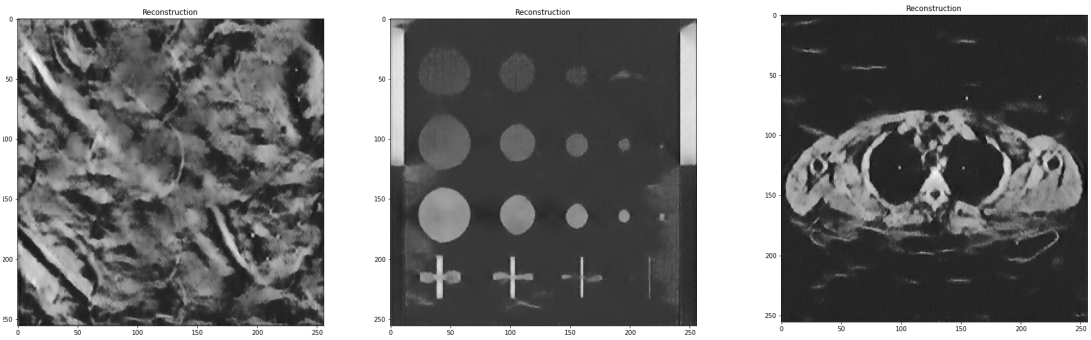


(c)

Figura 4.14: Training T3, 500 sintetiche, 300 reali, immagini corrotte su 100 angoli, a sinistra rete U-Net, a destra rete RED-CNN. In questo training tutte e tre le immagini sono migliorate su tutti gli aspetti, infatti sono più visibili i puntini e le masse. Si può notare che la rete U-Net restituisce dei risultati migliori rispetto alla rete RED-CNN.

4.4.3 Terzo test

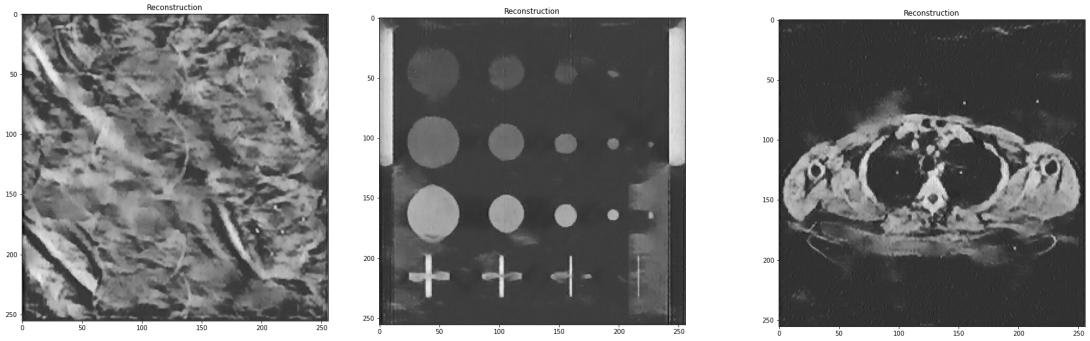
Con quest'ultimo training si ha avuto un miglioramento dell'immagine, che sembra più reale e conserva meglio i dettagli. In tutti questi esempi possiamo notare che la rete U-Net restituisce risultati migliori e ha anche un tempo di training inferiori. Perciò verranno effettuate ulteriore prove per cercare di migliorare la rete esclusivamente su quella U-Net.
La prima prova consiste nell'inserire l'activation *tanh* in tutti i livelli del decoder, che restituisce il seguente risultato.



(a)

Figura 4.15: Training T3, 500 sintetiche, 300 reali, immagini corrotte su 100 angoli, rete U-Net, activation tanh. I nuovi risultati sono visibilmente migliorati rispetto a prima, si riescono a distinguere bene i puntini e le masse.

Ora proveremo a cambiare tutti gli activation *tanh* in *relu*, tranne per l'ultimo livello del decoder che lo lasceremo *tanh*.



(a)

Figura 4.16: Training T3, 500 sintetiche, 300 reali, immagini corrotte su 100 angoli, rete U-Net, activation relu. Le immagini sono ancora migliorate, si vedono ancora meglio i puntini e le masse anche quelle con poco contrasto.

Come si può vedere da quest'ultimi risultati la rete che restituisce l'immagine migliore è quella con l'activation *relu*. Su questa rete calcoleremo l'intervallo di confidenza dell'MSE (Mean Squared Error)

Parametro	**Intervallo di confidenza**
MSE	da 0.01 a 0.02

Conclusioni

Per i test effettuati in questa tesi, sono state creati due dataset, uno con immagini sintetiche e uno con immagini reali prese dalla rete. Successivamente le due reti create venivano addestrate su questi dataset per poi verificare il training su alcune immagini test. Le immagini test servivano per verificare che la rete rilevasse i particolari importanti dell'immagine e per vedere il risultato in varie casistiche, come ad esempio la presenza di masse con contrasto diverso. Con il training su immagine sintetiche (figura 4.1, 4.2, 4.3) abbiamo potuto vedere che la rete U-Net restituiva risultati migliori rispetto alla RED-CNN, ma in ogni caso entrambe le reti riuscivano a rilevare i particolari interessanti e anche le masse con poco contrasto. Nel training su immagini reali (figura 4.7, 4.8, 4.9) si può notare che le immagini sintetiche venivano ricostruite male, spesso non erano preso dei dettagli, mentre l'immagine reale era buona, ma presentava del rumore al di fuori del soggetto dell'immagine. In questo training per le immagini sintetiche era meglio la ricostruzione della RED-CNN, mentre per l'immagine reale la U-Net. A seguito di questi risultati si è deciso di unire i due dataset, ma cercando di mantenere un certo rapporto tra i due tipi di immagini, altrimenti uno dei due dataset avrebbe influenzato troppo il risultato. Nella prima prova (figura 4.13) sono state prese 700 immagini sintetiche e 300 reali, il risultato è stato che le immagini sintetiche avevano perso i dettagli più piccoli e l'immagine finale presentava ancora del rumore. Dato questo risultato si è provato a ridurre le immagini sintetiche a 500 (figura 4.14), in questo test la rete U-Net ha restituito un buon risultato per tutte le immagini, sono stati rilevati anche i più dettagli nelle immagini sintetiche e l'immagine reale ha tutti i particolari e il rumore è praticamente scomparso. Una volta ottenuto il risultato ideale sono state effettuate due ulteriori prove per cercare di migliorare la rete, con la prima (figura 4.15) si ha avuto un leggero miglioramento, con la seconda (figura 4.16) si ha avuto un ulteriore miglioramento che è considerato il risultato migliore dei vari test.

Bibliografia

[1] *Deep Learning, cos'è l'apprendimento profondo, come funziona e quali sono i casi di applicazione:*
https://www.ai4business.it/intelligenza-artificiale/deep-learning/deep-learning-cose/

[2] *Semplice architettura di rete neurale convoluzionale:*
https://lorenzogovoni.com/architettura-di-rete-neurale-convoluzionale/

[3] *Le Reti Neurali Convoluzionali, ovvero come insegnare alle macchine a riconoscere per astrazione*
https://www.spindox.it/it/blog/reti-neurali-convoluzionali-il-deep-learning-ispir

[4] *An Intuitive Explanation of Convolutional Neural Networks*
https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

[5] Dottorato di Elena Morotti:
Reconstruction of 3D X-ray tomographic images from sparse data with TV-based methods

[6] *Low-Dose CT With a Residual Encoder-Decoder Convolutional Neural Network*
Di Hu Chen, Yi Zhang, Member, IEEE , Mannudeep K. Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, Senior Member, IEEE , and Ge Wang, Fellow, IEEE

[7] *U-Net FCN Networks | Deep Learning Engineer Italia*
https://andreaprovino.it/u-net/