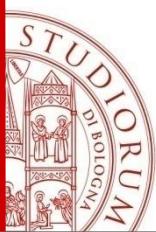


Blind deconvolution

Confronto tra metodi model-based e approcci deep learning supervisionati

Studente: Fabio Viggiano
Esame: Computational Imaging
Matricola: 0001195183



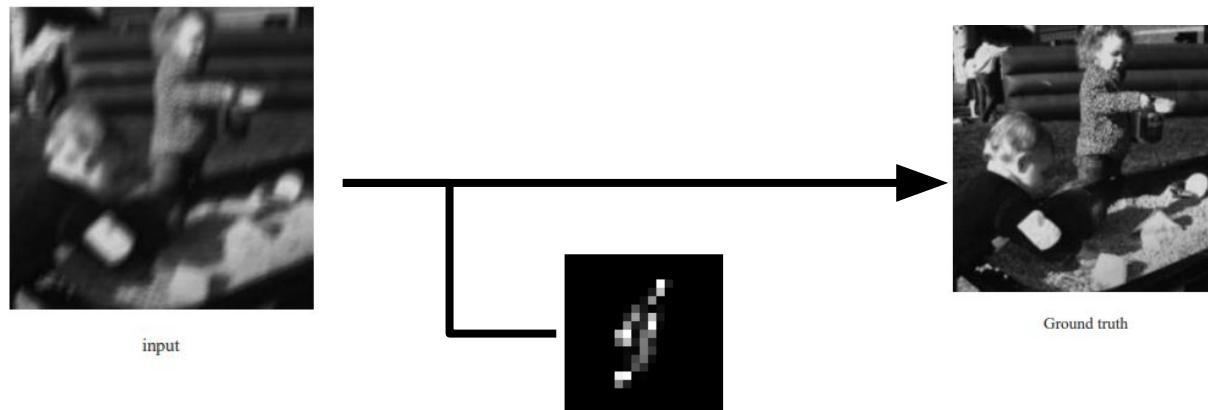
Il problema della blind deconvolution

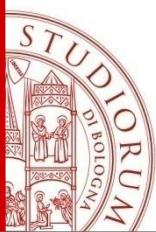
"Recuperare un'immagine nitida da una sua versione sfocata, senza conoscere a priori i dettagli esatti della sfocatura"

$$y = k \otimes x$$

Obiettivo: stimare x e k

- y : L'immagine sfocata
- k : Il kernel di sfocatura (blur) sconosciuto
- x : L'immagine nitida latente che vogliamo recuperare.





L'approccio standard: il metodo MAP

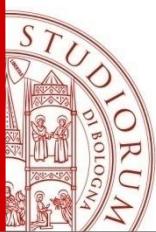
Idea di base: L'obiettivo è trovare la coppia (immagine x, kernel k) statisticamente più probabile.

$$\hat{x}, \hat{k} = \arg \max_{x,k} P(x, k|y)$$

Come: Si introduce un "prior" sulle immagini naturali (gradienti sparsi). L'aspettativa è che questo favorisca la ricostruzione dell'immagine nitida.

Il problema: Contrariamente alle attese, questo approccio tende a favorire soluzioni "banali" (no-blur).

Perché?: Un'immagine sfocata ha gradienti più bassi e meno texture, risultando per il modello "statisticamente più semplice" e quindi più probabile.



La prova visiva del fallimento del MAP (Levin et al.)

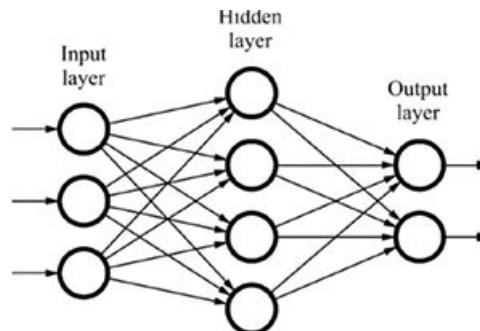


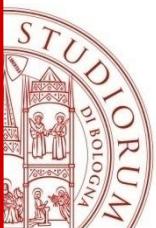
Levin et al. dimostrano questo fallimento analizzando un'immagine a piccole porzioni. Le aree in rosso sono le uniche (il 3%) in cui il modello favorisce la versione nitida. Nel restante **97% dei casi, la versione sfocata è considerata statisticamente "migliore"**.

Questo giustifica la necessità di approcci più sofisticati.

Le soluzioni proposte

Due approcci

Model-based (Shan et al.)	Data-driven (U-Net)
<p>Si basa su un modello matematico esplicito dell'immagine e del processo di sfocatura. Cerca una soluzione tramite un ottimizzazione iterativa.</p> <p> Input sfocato →  Stima kernel →  Immagine ricostruita</p>	<p>Non usa un modello esplicito, ma "impara" a rimuovere il blur attraverso l'osservazione di migliaia di esempi (addestramento supervisionato).</p> 



Approccio #1

L'Algoritmo di Shan (2008)

Questo metodo non si affida a un approccio MAP standard, ma lo migliora attraverso una strategia di ottimizzazione più sofisticata.

L'obiettivo è minimizzare una funzione di costo che bilancia due termini:

- **Fedeltà ai dati:** Quanto l'immagine ricostruita, una volta sfocata con il kernel stimato, assomiglia all' input.
- **Prior dell'immagine:** Quanto "naturale" appare l'immagine ricostruita (favorendo gradienti sparsi).

L'Algoritmo di Shan (2008)

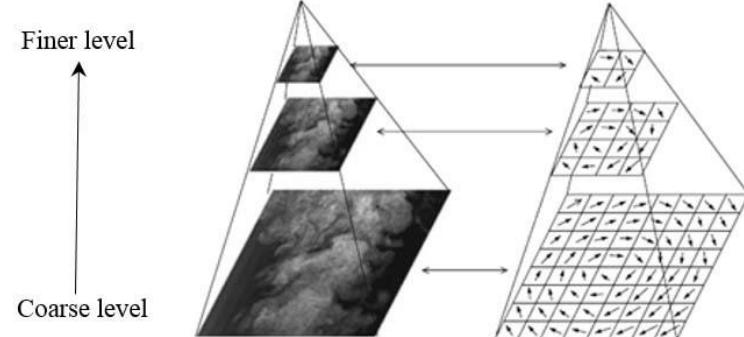
I suoi punti di forza e le sue caratteristiche principali sono:

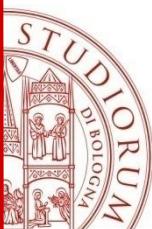
Strategia piramidale (Coarse-to-Fine)

- L'algoritmo non lavora subito sull'immagine a piena risoluzione.

Prima stima il kernel su una versione rimpicciolita dell'immagine per ottenere una stima "grossolana" ma robusta del movimento principale.

- Successivamente, rifinisce questa stima su versioni via via più grandi dell'immagine. Questo approccio lo aiuta a evitare di convergere a soluzioni errate.





L'Algoritmo di Shan (2008)

Ottimizzazione alternata

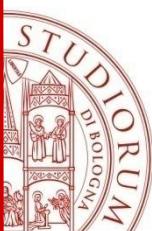
Il problema di trovare l'immagine nitida (x) e il kernel (k) contemporaneamente è troppo complesso.

L'algoritmo lo risolve alternando due passi in modo iterativo:

- Stima dell'immagine: Tenendo fisso il kernel k , trova la migliore immagine nitida x .
- Stima del kernel: Tenendo fissa la nuova immagine x , trova un kernel k migliore.
- Questi due passi vengono ripetuti *fino a convergenza*.

Processo:

Input → Coarse-to-Fine → Alternating Optimization → Output



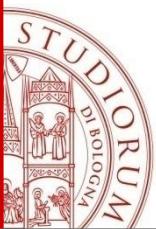
Shan et al. (2008)

Criticità emerse dalla sperimentazione

Dalle analisi è emerso che l'algoritmo è estremamente sensibile all'
intensità del blur.

Come vedremo nei risultati, le sue performance crollano rapidamente
su *immagini molto sfocate*.

È inoltre molto sensibile alla scelta dei parametri (es. `kernel_size`),
richiedendo un'attenta calibrazione per ogni scenario

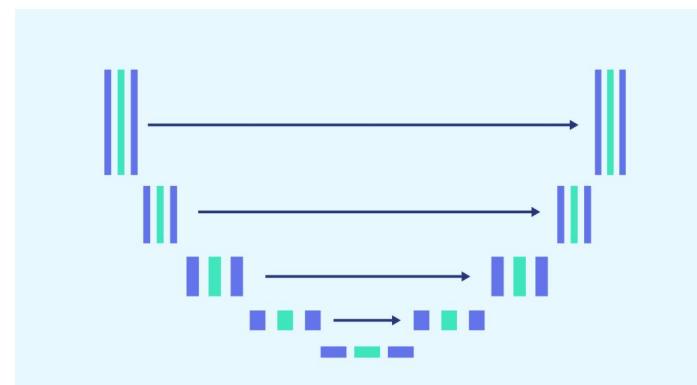


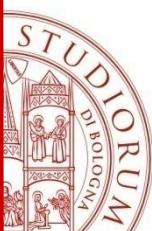
Approccio #2

La rete neurale U-Net

A differenza di un approccio model-based, questo metodo non si basa su un modello matematico esplicito del blur.

Al contrario, **impara a risolvere il problema osservando migliaia di esempi.**





La Rete Neurale U-Net

Concetto: **l'addestramento supervisionato**

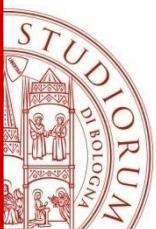
Si parte da un vasto *dataset di coppie di immagini*: (immagine sfocata, immagine nitida corrispondente).

La rete neurale prende *in input un'immagine sfocata e produce una sua stima dell'immagine nitida*.

Questa stima viene confrontata con l'immagine nitida reale (il "ground truth") per calcolare un "errore" (loss).

L'errore viene usato per aggiornare i parametri interni della rete, in un processo che mira a minimizzare l'errore stesso.

Ripetendo questo processo per migliaia di volte, la rete "impara" la funzione che mappa un'immagine sfocata alla sua versione nitida.

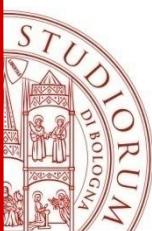


La rete neurale U-Net

Il contributo pratico: la creazione del dataset

Per addestrare la rete, è stato generato un dataset di circa 10.000 coppie di immagini, utilizzando lo script `create_dataset.py`.

Sono stati applicati blur sintetici con parametri scelti casualmente per garantire una buona varietà di esempi.



Approccio #2 - La Rete Neurale U-Net

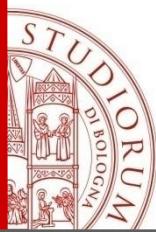
Architettura: perché la U-Net?

Abbiamo scelto una U-Net, un architettura molto efficace per compiti image-to-image. La sua forza risiede nella sua struttura simmetrica:

Encoder (percorso di contrazione): Analizza l'immagine di input e ne estrae le caratteristiche a diverse scale, catturando il contesto.

Decoder (percorso di espansione): Ricostruisce l'immagine nitida partendo dalle caratteristiche apprese.

Skip Connections: Il suo elemento chiave. Collegano direttamente i layer dell'encoder a quelli del decoder, permettendo alla rete di non perdere i dettagli fini dell'immagine originale durante il processo di ricostruzione.

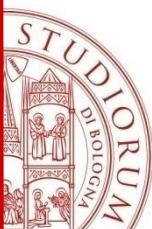


Setup sperimentale

Setup sperimentale: come abbiamo misurato le performance

Dataset di riferimento:

"Per tutti i nostri test, siamo partiti da un set di immagini nitide di base (principalmente dal dataset di immagini mediche C081)."



Setup Sperimentale

Generazione controllata del Blur

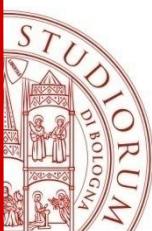
Per poter misurare l'errore in modo oggettivo, abbiamo applicato un blur sintetico alle immagini nitide, creando così le nostre immagini di input e conoscendo il risultato perfetto (ground truth).

"Abbiamo testato due tipi di blur, rilevanti nel mondo reale:"

Motion Blur: Simula il mosso dovuto al *movimento della fotocamera*.

Gaussian Blur: Simula una sfocatura più generale, come quella dovuta a una *messa a fuoco non perfetta*.

Per ciascun tipo, abbiamo testato diversi livelli di intensità:
leggero, moderato e intenso.



Metriche di valutazione

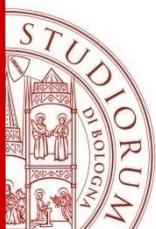
Per confrontare i risultati in modo quantitativo, abbiamo usato due metriche standard:

PSNR (Peak Signal-to-Noise Ratio): Misura l'errore tra l'immagine ricostruita e quella originale. È misurato in decibel (**dB**).

Più alto è il valore, migliore è la ricostruzione.

SSIM (Structural Similarity Index Measure): Misura la somiglianza percepita tra le due immagini, considerando struttura, luminosità e contrasto. Il valore va da 0 a 1.

Più vicino a 1 è il valore, migliore è la ricostruzione.

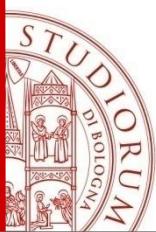


Risultati Quantitativi - Tabella Comparativa

Metodo	Livello Blur	PSNR (dB) ↑	SSIM ↑
Shan	Realistico	23.32	0.6800
Shan	Forte	12.92	0.600
U-Net	Vario	34.18	0.9056

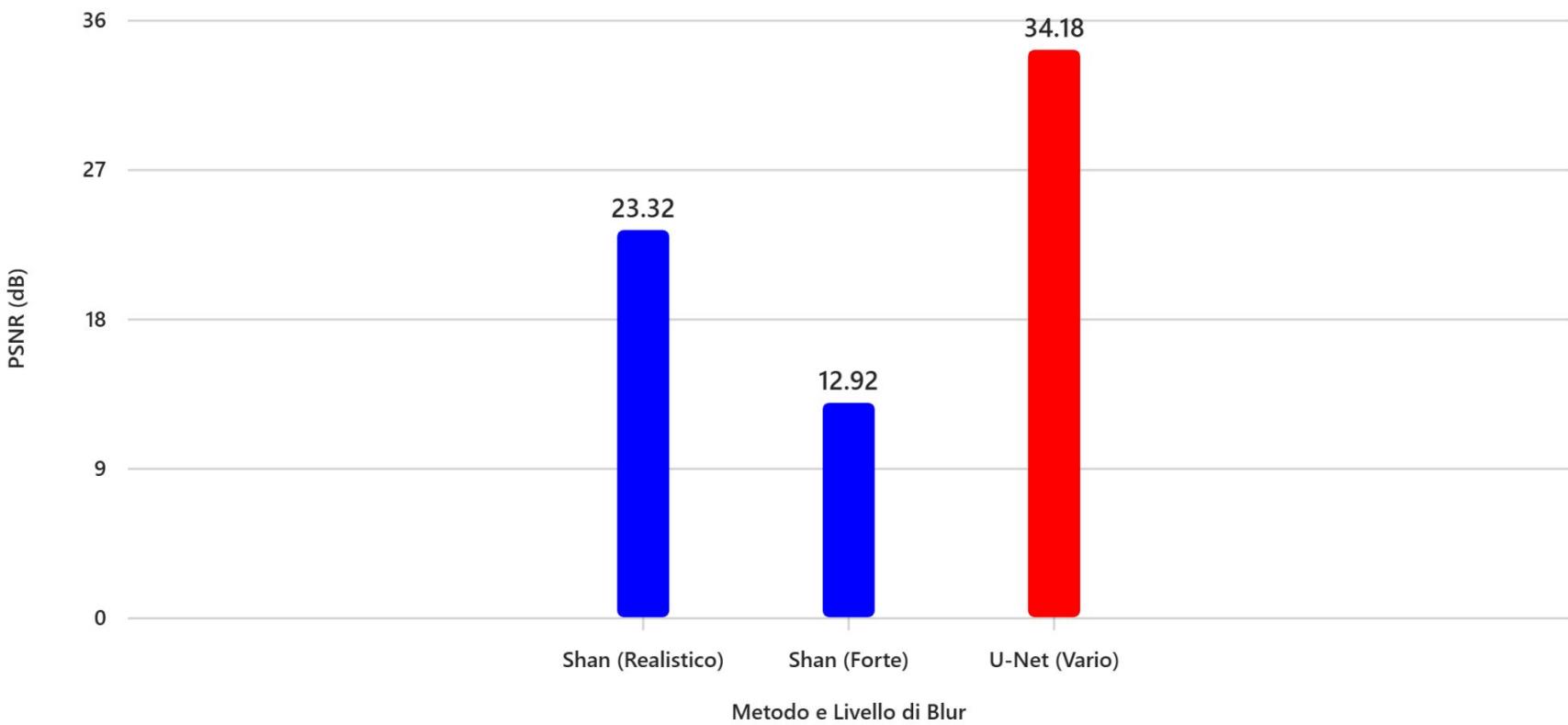
I valori mostrano chiaramente:

- **Shan**: degrado rapido con blur intenso.
- **U-Net**: mantiene PSNR e SSIM elevati anche con blur forte.

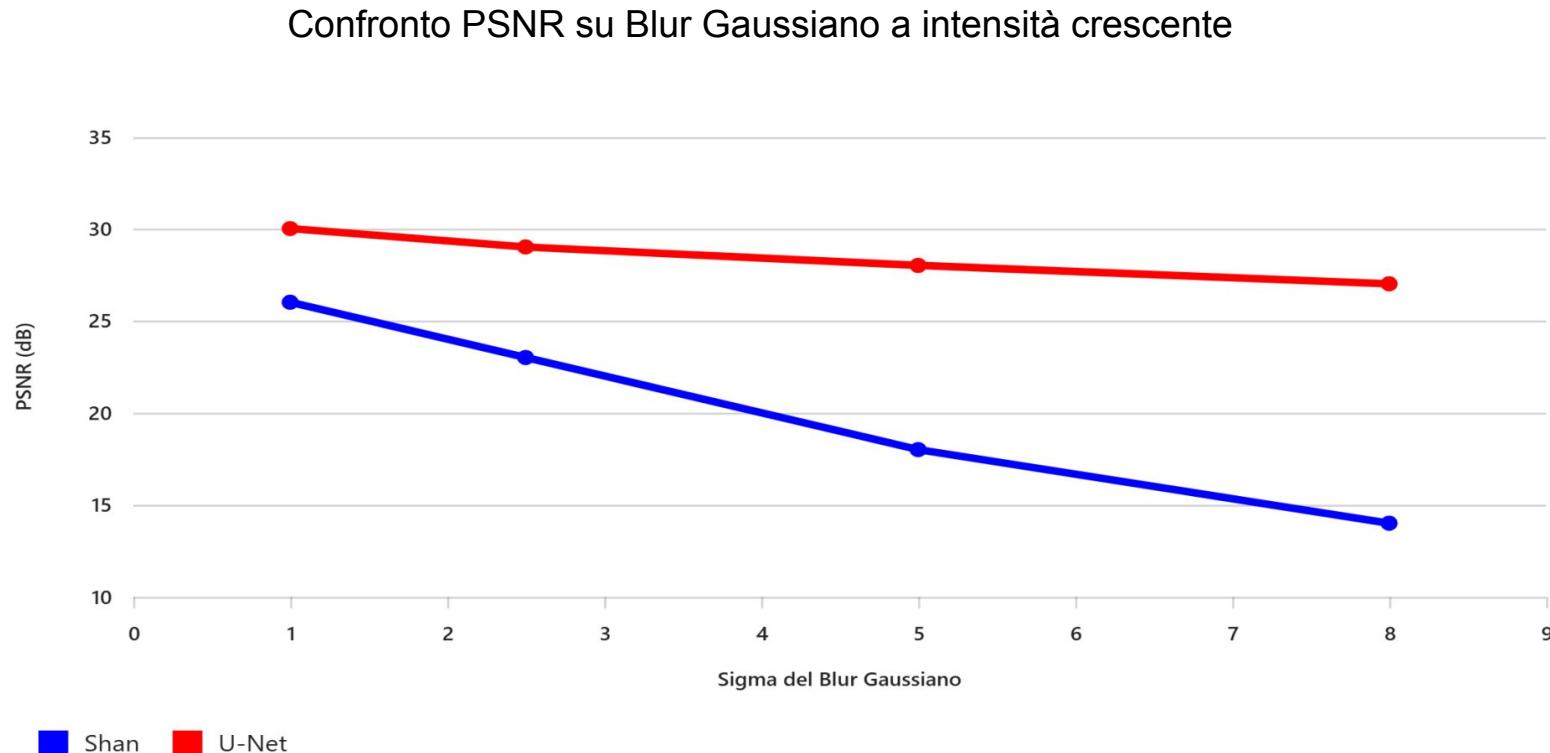


Analisi grafica: performance al variare dell'intensità del blur

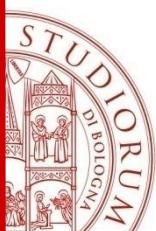
Confronto PSNR su Blur Gaussiano a intensità crescente



Analisi grafica: performance al variare dell'intensità del blur



La linea blu mostra la sensibilità di Shan all'aumento di sigma, mentre la linea rossa di U-Net resta stabile.



Risultati qualitativi: un confronto visivo | Best case

Immagine ricostruita
27_v2_deblurred

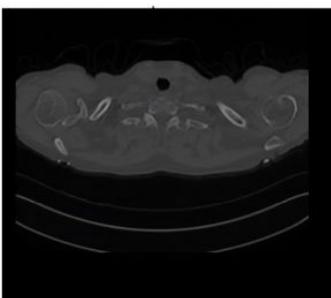


Immagine Sfocata (Input)

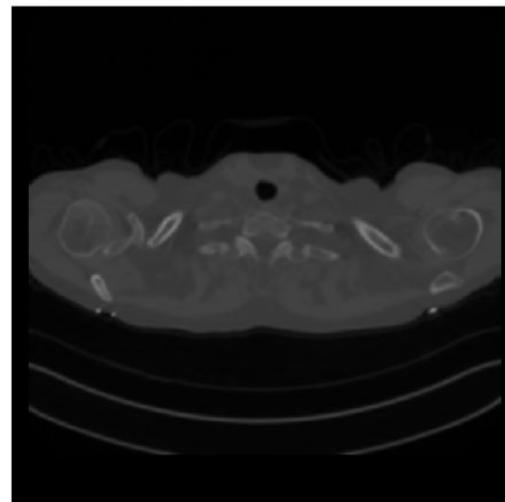
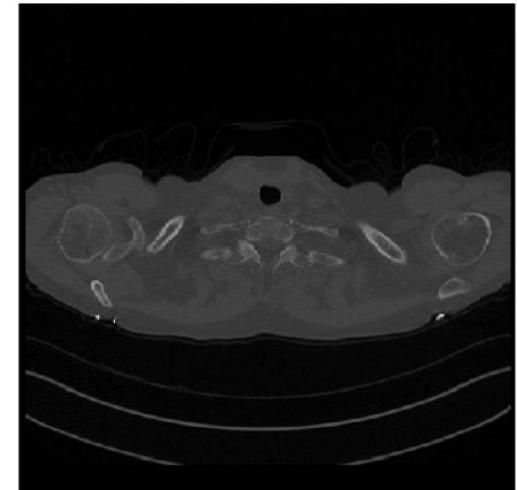


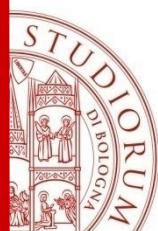
Immagine Nitida (Originale)



File: 27_v2.png

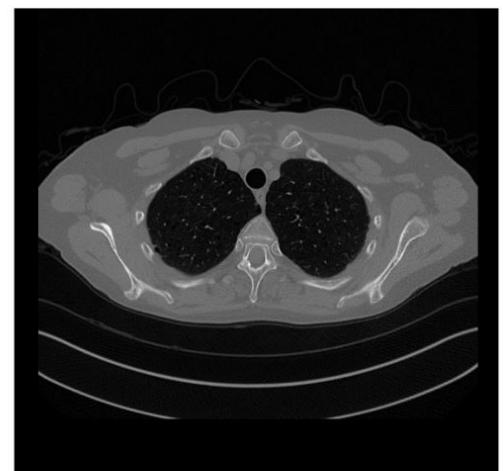
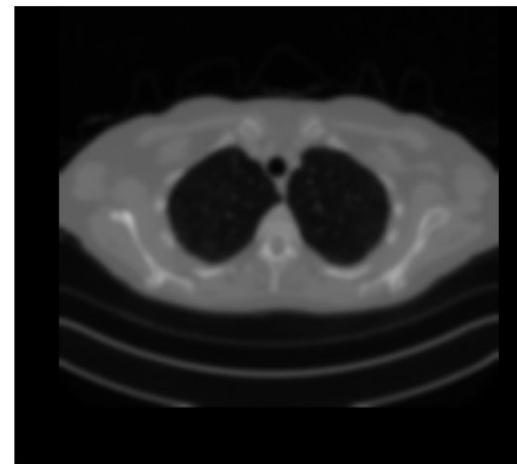
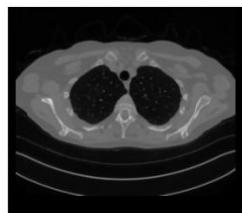
Metriche: PSNR: 40.34 dB, SSIM: 0.9645

Ha uno dei valori di PSNR e SSIM più alti dell'intero test set. Dimostra quanto è accurata la ricostruzione quando le condizioni sono favorevoli.



Risultati qualitativi: un confronto visivo | Average case

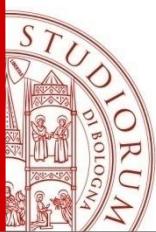
Immagine Ricostruita (Output): results_nn/103_v0_deblurred.png - Immagine Sfocata (Input): data_processed/test/blurred/C081/103_v0.png - Immagine Nitida (Originale): data_processed/test/sharp/C081/103_v0.png



File: 103_v0.png

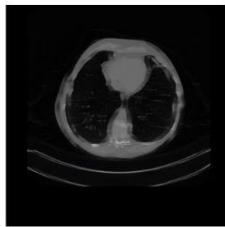
Metriche: PSNR: 34.40 dB, SSIM: 0.9177

Il suo valore di PSNR è quasi identico alla media generale, rappresenta perfettamente un risultato "standard" del modello.



Risultati qualitativi: un confronto visivo | Worst case

Immagine Ricostruita (Output): results_nn/222_v0_deblurred.png



File: 222_v0.png

Metriche: PSNR: 22.53 dB, SSIM: 0.7258

Ha uno dei valori di PSNR e SSIM più bassi di tutto il set. La ricostruzione è visibilmente imperfetta.

Immagine Sfocata (Input): data_processed/test/blurred/C081/222_v0.png

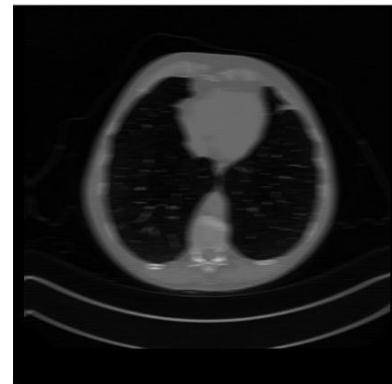
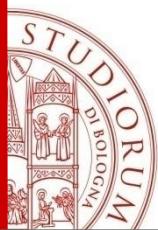


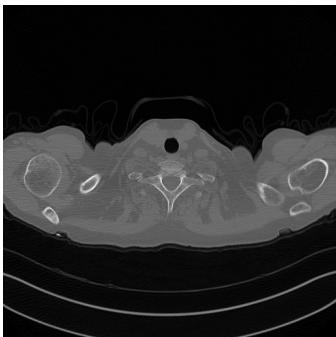
Immagine Nitida (Originale): data_processed/test/sharp/C081/222_v0.png



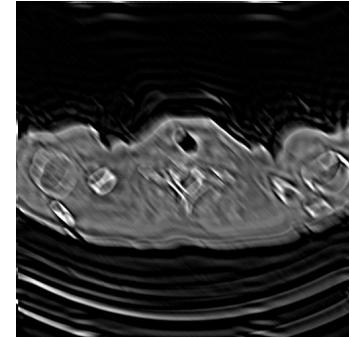


Confronto visivo tra Shan e U-Net

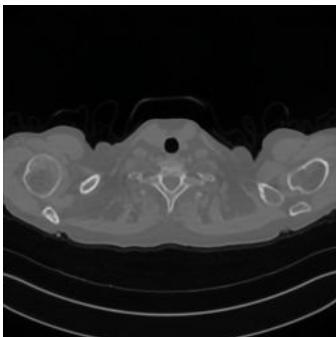
Caso rappresentativo



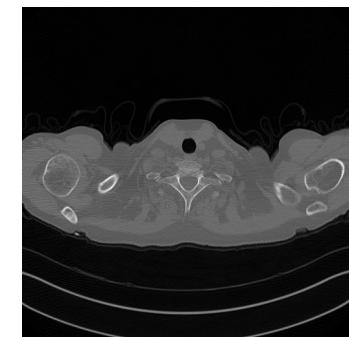
Input sfocato : data/test/C081/15.png



Output Shan: PSNR = 18.21, SSIM = 0.3758.

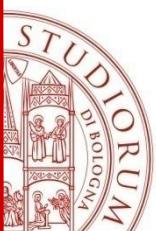


Output U-Net: 103_v0.png PSNR ≈ 34.40, SSIM ≈ 0.9177



Ground Truth: test/C081/15.png

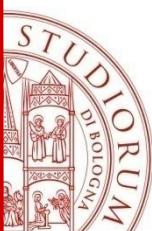
U-Net recupera più dettagli e struttura rispetto a Shan, soprattutto in presenza di blur moderato



Risultati chiave della sperimentazione

L'analisi comparativa ha dimostrato che, sebbene entrambi gli approcci siano validi, *l'approccio data-driven basato su rete neurale (U-Net) supera nettamente il metodo model-based (Shan et al.).*

La superiorità della U-Net è particolarmente evidente in condizioni di blur intenso, dove l'algoritmo classico mostra i suoi limiti, mentre la rete neurale dimostra una notevole robustezza.

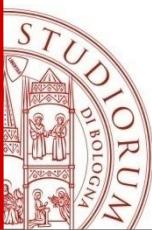


Conclusioni e prospettive future

Un confronto tra due filosofie (Trade-offs):

Approccio Model-Based (Shan et al.):

- **Pro:** Non richiede un dataset di training. È basato su principi matematici esplicativi e interpretabili.
- **Contro:** Meno performante, più lento in fase di esecuzione (inferenza) e molto sensibile ai parametri.



Conclusioni e prospettive future

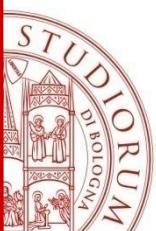
Un confronto tra due filosofie (Trade-offs):

Approccio

Data-Driven

(U-Net):

- **Pro:** Performance nettamente superiori, specialmente nei casi difficili. Una volta addestrata, è estremamente veloce nell'elaborare nuove immagini.
- **Contro:** Richiede un grande sforzo iniziale per la creazione del dataset e un lungo processo di addestramento (computazionalmente costoso). Opera come una "black box", rendendo più difficile interpretare le sue decisioni.



E Fergus?

L'osservazione

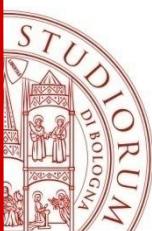
di

Fergus

Le immagini naturali non sono casuali.

Se calcoliamo i **gradienti** (le differenze tra pixel adiacenti), la loro distribuzione ha una *forma molto particolare*: tantissimi valori vicini a zero (le aree uniformi come il cielo o un muro) e pochi valori molto alti (i bordi netti, i contorni).

Ha modellato questa distribuzione con una **distribuzione a coda pesante** (heavy-tailed).



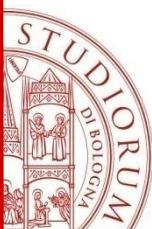
E Fergus?

Per trovare la L e la K che massimizzano questa probabilità:

Variational Bayes o **Expectation-Maximization (EM)**.

L' algoritmo itera tra due passi principali:

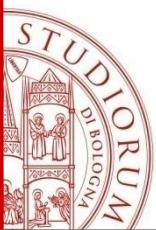
- E-Step (Estimation): Stima la **distribuzione di probabilità dell'immagine latente L** , dato il kernel K attuale. Questo passo è una forma di deconvolution non-blind, ma guidata dal prior statistico sull'immagine.
- M-Step (Maximization): Aggiorna la **stima del kernel K per massimizzare la probabilità**, data la distribuzione dell'immagine L appena stimata



Shan vs Fergus

Shan: Usa una regolarizzazione che penalizza i gradienti per favorire la "sparsità" (poche aree non uniformi).

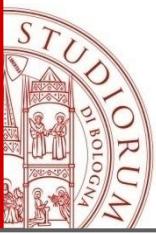
Fergus: Usa un modello statistico **esplicito** della distribuzione dei gradienti delle immagini natura. Questo modello "sa" che le immagini sono fatte di vaste aree piatte *ma anche* di alcuni bordi molto netti. Di conseguenza, è molto più bravo a preservare i contorni importanti senza sacrificare le texture.



Implicazioni e prospettive future

Per problemi complessi di inverse imaging come la blind deconvolution, gli approcci supervisionati, pur richiedendo un investimento iniziale maggiore, si dimostrano più efficaci e generalizzabili nel mondo reale.

Sviluppi futuri potrebbero includere l'addestramento su dataset con blur non uniformi per simulare movimenti più realistici, o l'esplorazione di architetture di rete ancora più avanzate.



Grazie per l'attenzione