



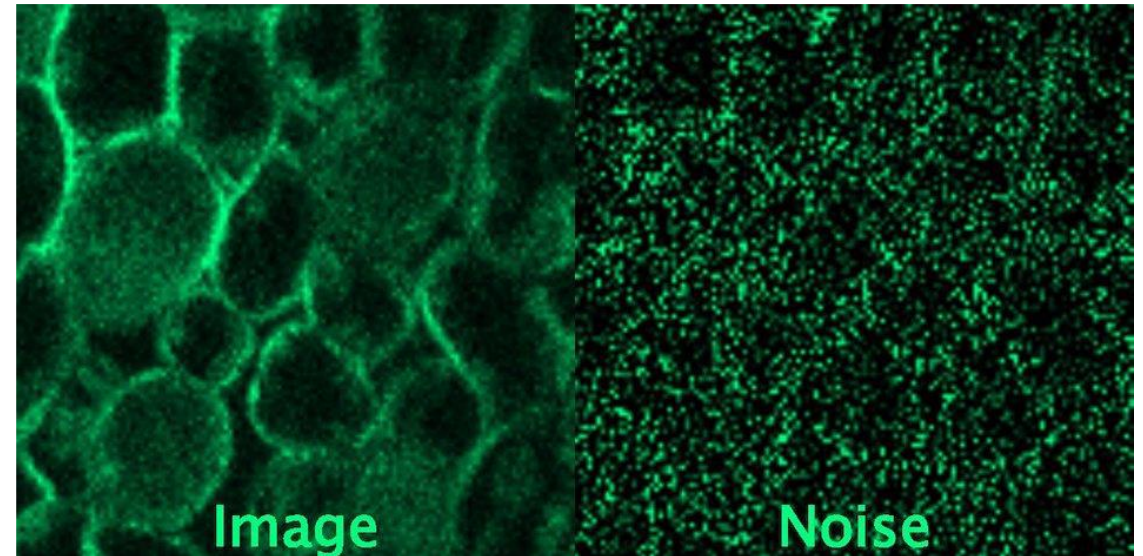
Denoising basato su deep learning

Ranzoni Valentina

Paolo Parati

Cos'è il Rumore:

- **Distorsione casuale** che altera i pixel, riducendo la chiarezza dell'immagine.
- **Artefatto** che può influenzare l'analisi dei dati.
- Compromette **nitidezza, contrasto e dettagli** dell'immagine.



Come si forma il rumore

- **Rumore da sensore:** I sensori non catturano tutte le informazioni luminose.
- **Interferenze elettroniche:** Disturbi nei circuiti elettronici che alterano i segnali.
- **Limitazioni hardware:** Qualità inferiore dei dispositivi di acquisizione (bassa risoluzione).
- **Condizioni ambientali:** Fluttuazioni di temperatura, illuminazione instabile o vibrazioni.
- **Compressione dei dati:** Algoritmi di compressione che riducono la dimensione del file ma causano artefatti.



Classificazione del rumore

Rumore Strutturato: configurazione regolare o pattern riconoscibile, facilmente identificabile.

- *Rumore Sale e Pepe:* pixel bianchi e neri casuali sparsi nell'immagine.
- *Rumore a Bande* (Banding Noise): linee orizzontali o verticali regolari nell'immagine.

Rumore Non Strutturato: completamente casuale e non segue alcun pattern.

- *Rumore Gaussiano:* casuale e segue una distribuzione normale (gaussiana).
- *Rumore di Poisson:* presente in immagini con scarsa illuminazione, presenta variazioni casuali di intensità.

Denoising di Immagini Digitali

- Ridurre il rumore senza compromettere alcune caratteristiche essenziali:
- Le **zone uniformi** devono restare omogenee
- I **bordi** devono essere preservati
- Le **texture** devono essere mantenute
- Non devono essere generati nuovi **artefatti**



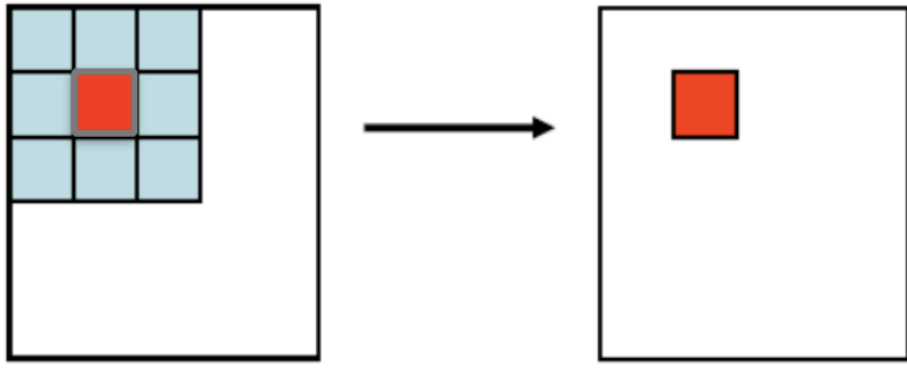
Approcci Principali al Denoising

- Filtraggio nel dominio spaziale
- Filtraggio nel dominio delle trasformazioni
- Metodi avanzati
- Metodi basati sull'apprendimento automatico



Filtraggio nel Dominio Spaziale

Definizione: Operazioni sui pixel sfruttando correlazioni locali.



Filtri lineari:

Filtro medio: Media dei pixel circostanti.

Filtro gaussiano: Ponderazione centrata.

Filtri non lineari:

Filtro mediano: Valore mediano dei pixel circostanti.

Filtro bilaterale: Combina distanza spaziale e intensità per preservare i bordi.

Filtraggio nel Dominio delle Trasformazioni

Processo:

- Trasformare l'immagine in un dominio alternativo.
- Applicare filtri per ridurre il rumore.
- Tornare al dominio spaziale.

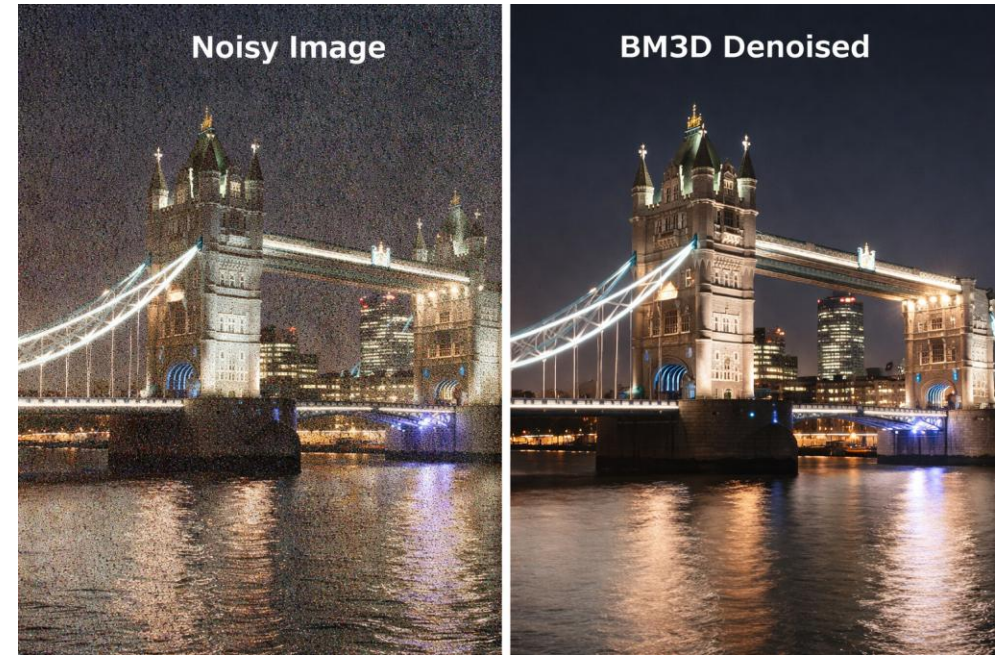
Tecniche principali:

- Trasformata di Fourier (FT): Rimozione delle alte frequenze.
- Trasformata wavelet (WT): Multi-scala per separare dettagli e rumore.
- Trasformazioni adattive: Si adattano alle caratteristiche specifiche dell'immagine e del rumore

Metodi Avanzati: BM3D

Adotta un approccio iterativo in due stadi costituiti dalle seguenti operazioni di base:

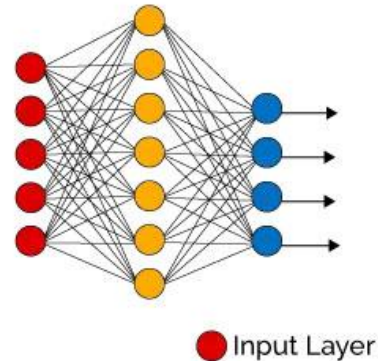
- Block Matching: Regioni simili nell'immagine vengono raggruppate.
- Collaborative Filtering: Applicazione di filtri nel dominio wavelet.
- Aggregation: Combina blocchi ripuliti.



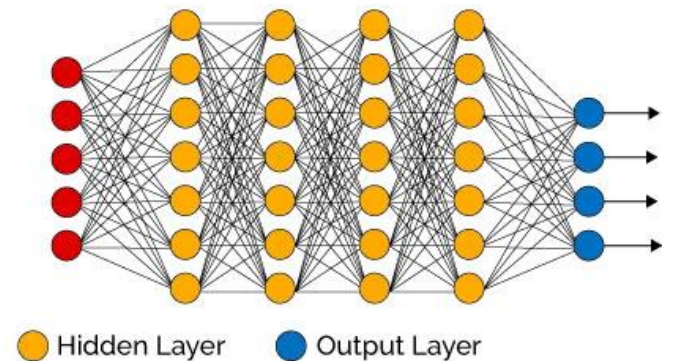
Limiti dei metodi classici

- I metodi classici di denoising (es. BM3D) si basano su ipotesi predefinite e parametri manuali
- Scarsa adattabilità a rumori complessi o non stazionari
- Limitata preservazione di dettagli fini e texture, soprattutto con rumore elevato
- Costo computazionale elevato, poco adatti al real-time
- Questi limiti favoriscono l'uso di CNN, che apprendono dal dato modelli di rumore più flessibili ed efficaci

Rete neurale semplice



Rete neurale Deep Learning

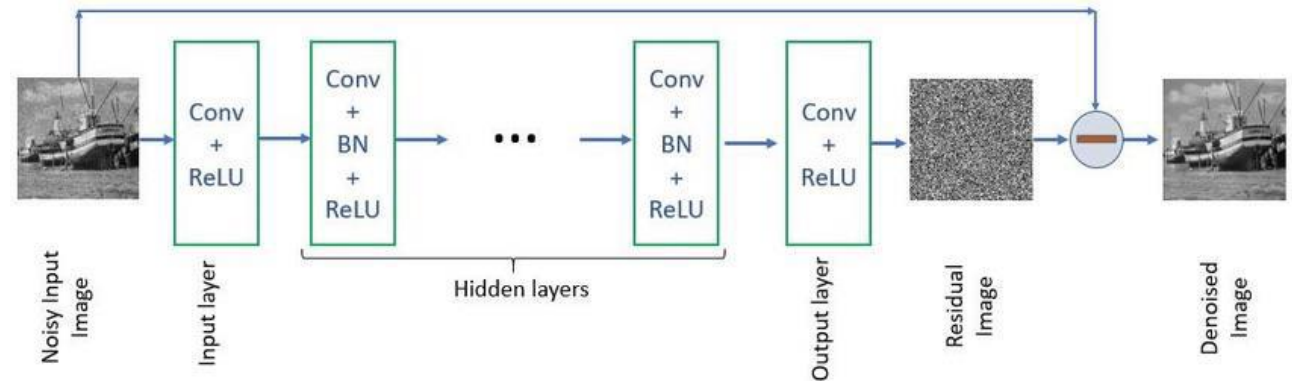


Apprendimento Automatico

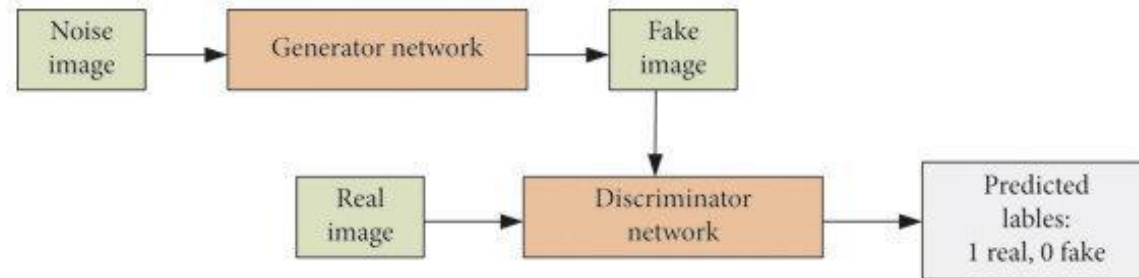
1. Convolutional Neural Networks (CNN)

Architettura DnCNN: predice il rumore, lo sottrae per ottenere l'immagine pulita.

- Input: immagine rumorosa
- Output previsto: rumore sintetico aggiunto
- Loss Function: differenza tra il rumore predetto dalla rete e il rumore aggiunto



Apprendimento Automatico



2. Generative Adversarial Networks (GAN)

- Due reti neurali che lavorano in competizione tra loro:
- Generatore: produce immagini pulite che siano il più simili possibile a quelle reali.
- Discriminatore: valuta l'immagine e cerca di distinguere tra immagini reali e false.

Ground Truth

Cos'è la ground truth:

La **ground truth** è il dato di riferimento considerato corretto, che rappresenta l'informazione reale attesa e viene utilizzato per addestrare e valutare le prestazioni di un algoritmo.

In molti contesti reali non è disponibile o non acquisibile

Ground Truth

Per superare la mancanza di ground truth, sono stati introdotti approcci di **apprendimento self-supervised**, che apprendono direttamente dalle immagini rumorose senza necessità di dati puliti. Un esempio è **Noise2Void**, che sfrutta la ridondanza spaziale dell'immagine per predire il valore di un pixel a partire dal suo intorno, consentendo il denoising senza immagini di riferimento.

Approcci Innovativi

Noise2Noise (N2N):

- Modello allenato su due immagini rumorose della stessa scena.
- Immagini basate sullo stesso contenuto che sono "rumorose" in modi diversi
- Vantaggio: Non richiede immagini pulite.

Noise2Void (N2V):

- Lavora su una sola immagine rumorosa.
- Vantaggio: Perfetto per contesti con dati limitati.

Challenge

Abbiamo deciso di condurre un confronto tra 2 dataset per valutare l'efficacia della strategia di denoising basata su U-Net e Noise2Void.

In questo caso permette di analizzare il comportamento del metodo in due scenari differenti:

- uno controllato
 - con immagini naturali affette da rumore artificiale
- uno realistico, con immagini di microscopia biologica caratterizzate da rumore reale e complesso.

Nel primo caso, la disponibilità del ground truth permette una valutazione quantitativa, verificando la correttezza del denoising.

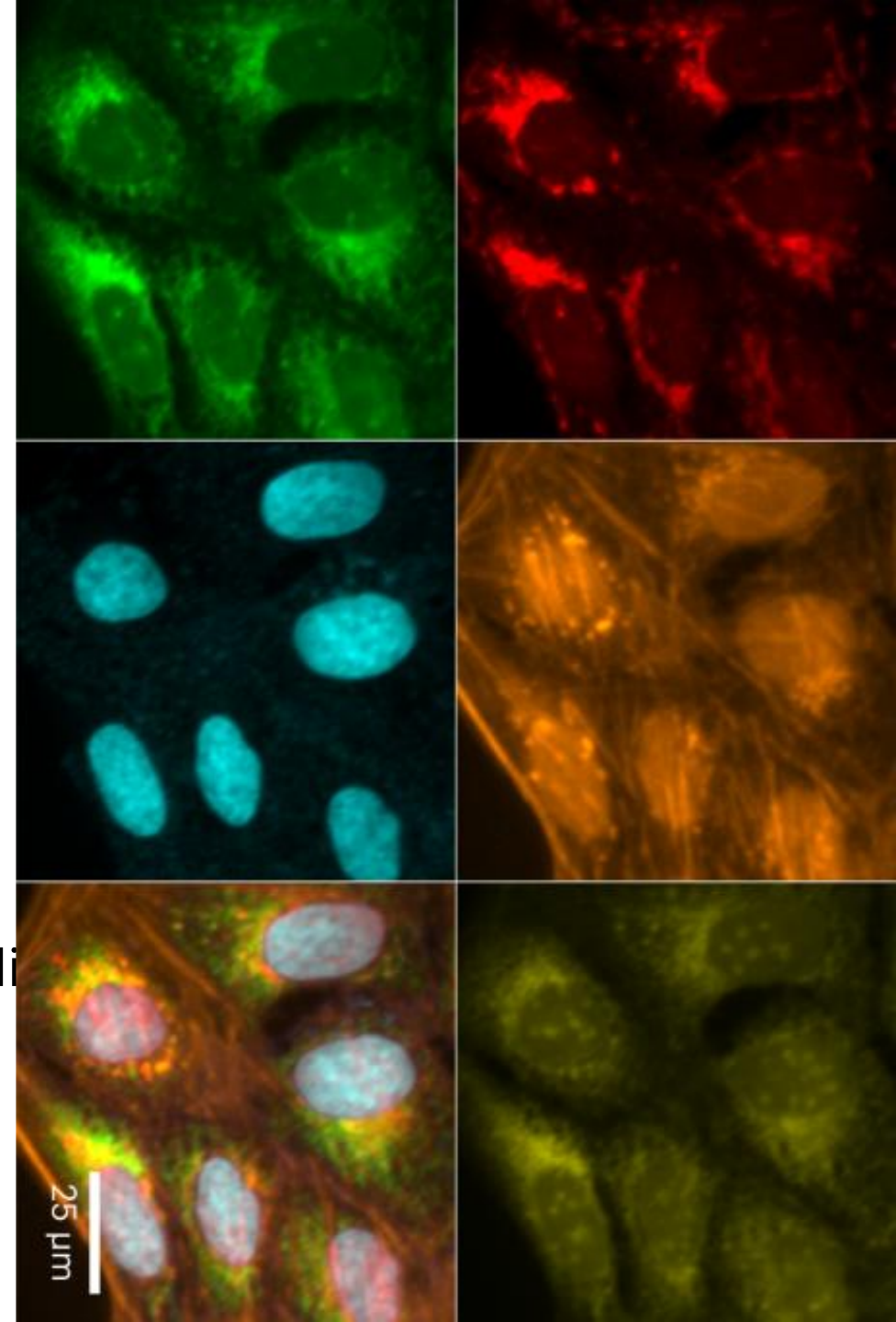
Nel secondo caso, l'analisi è principalmente qualitativa e orientata alla preservazione delle strutture biologiche, permettendo di valutare l'affidabilità del metodo in un contesto applicativo reale.

Jump Dataset

- immagini di microscopia cellulare
 - Il dataset è ampiamente impiegato in ambito biomedico
 - Caratterizzato da immagini acquisite in condizioni reali
 - Non c'è una versione pulita di riferimento
- Rumore di acquisizione
- Limitazione del sensore
 - Condizione di illuminazione

Abbiamo scelto questo dataset per l'assenza di una ground truth pulita-

Rende adatta l'utilizzo di approcci self-supervised come Noise2Void

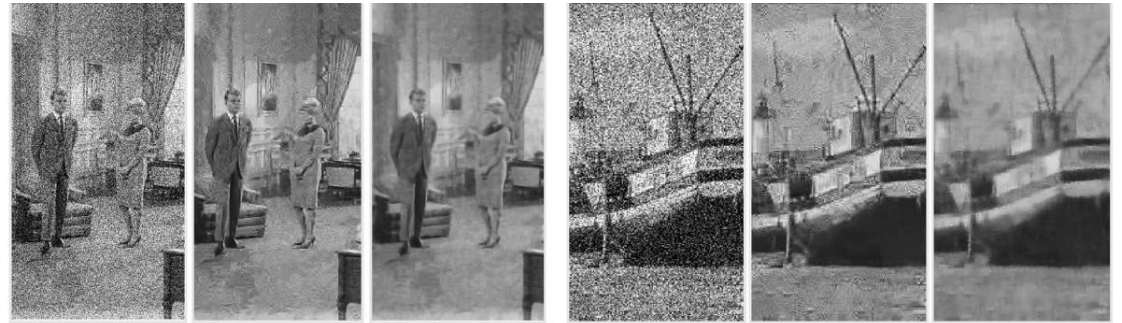


BSD68

- 68 immagini naturali con rumore artificiale
- **Rumore artificiale = esperimento controllato**

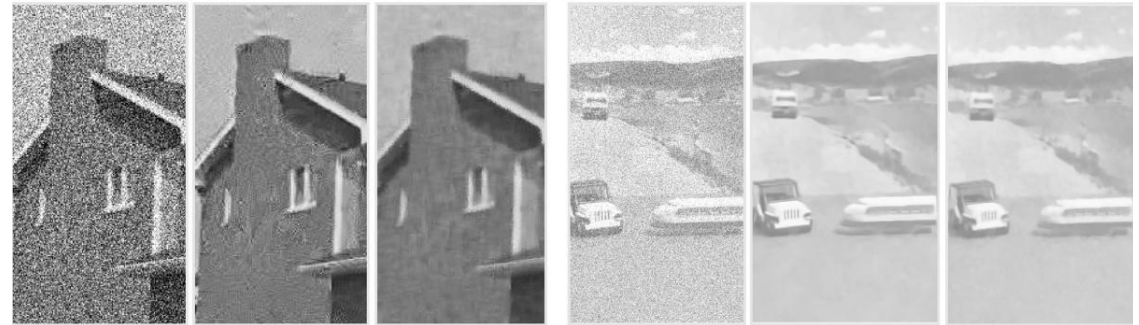
Utile perché fornisce un contesto standard e controllato che rende possibile confrontare in modo oggettivo metodi di denoising tramite metriche come

- PSNR
- SSIM



(a)

(b)



(c)

(d)

BSD68

PSNR (Peak Signal-to-Noise Ratio)

- misura **l'errore pixel-per-pixel**
- più è alto → più l'immagine denoised è vicina alla ground truth

SSIM (Structural Similarity Index)

- misura quanto sono simili **strutture, forme e contrasto**
- più è vicino a **1**, meglio è

Preprocessing dei dati

Insieme di operazioni volte a trasformare i dati grezzi acquisiti in una rappresentazione che sia compatibile con il funzionamento di una rete neurale convoluzionale.

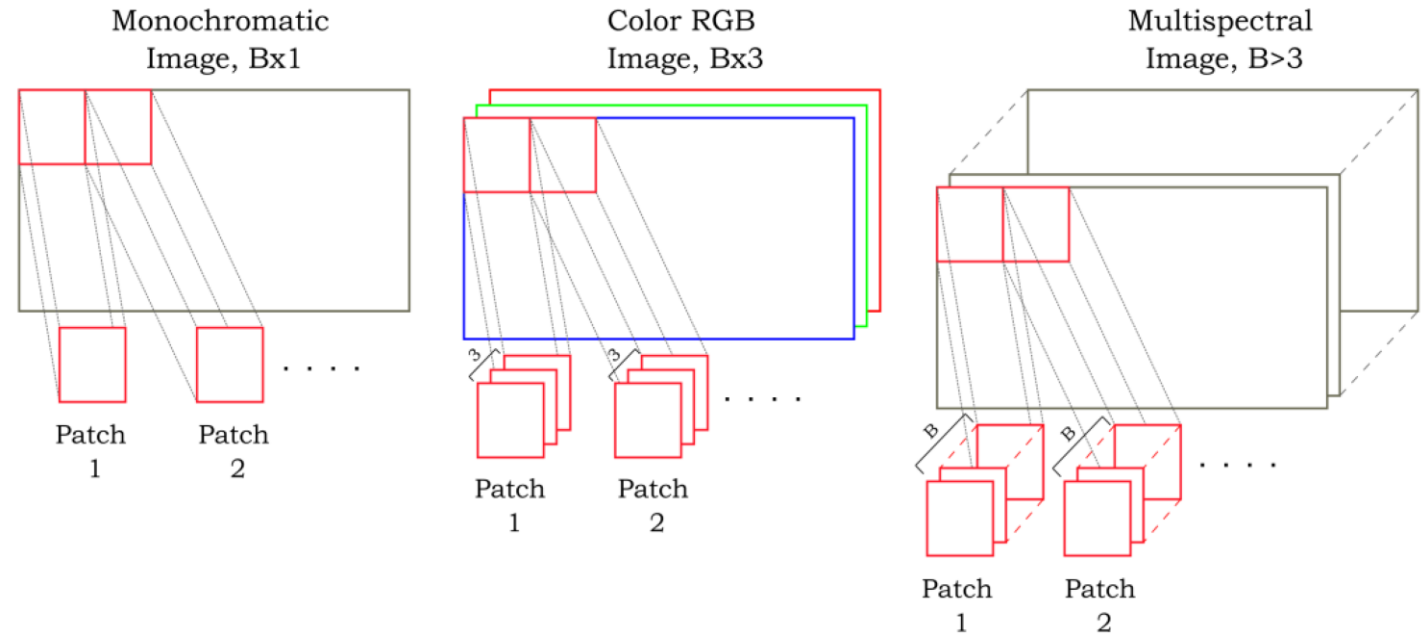
I valori dei pixel possono avere scale molto diverse tra canali e tra immagini differenti, questo può rendere instabile l'addestramento.

1. Normalizzazione

- si riportano i valori dei pixel in intervallo standard
- viene ridotta la variabilità artificiale tra immagini

Preprocessing dei dati

Suddivisione in patch di dimensione fissa



- Aumenta il numero di campione disponibili per l'addestramento
- Consente di concentrarsi su regioni locali dell'immagine dove il rumore è più omogeneo
- Riduce il carico computazionale

Noise2Void

Grazie al mascheramento dei pixel

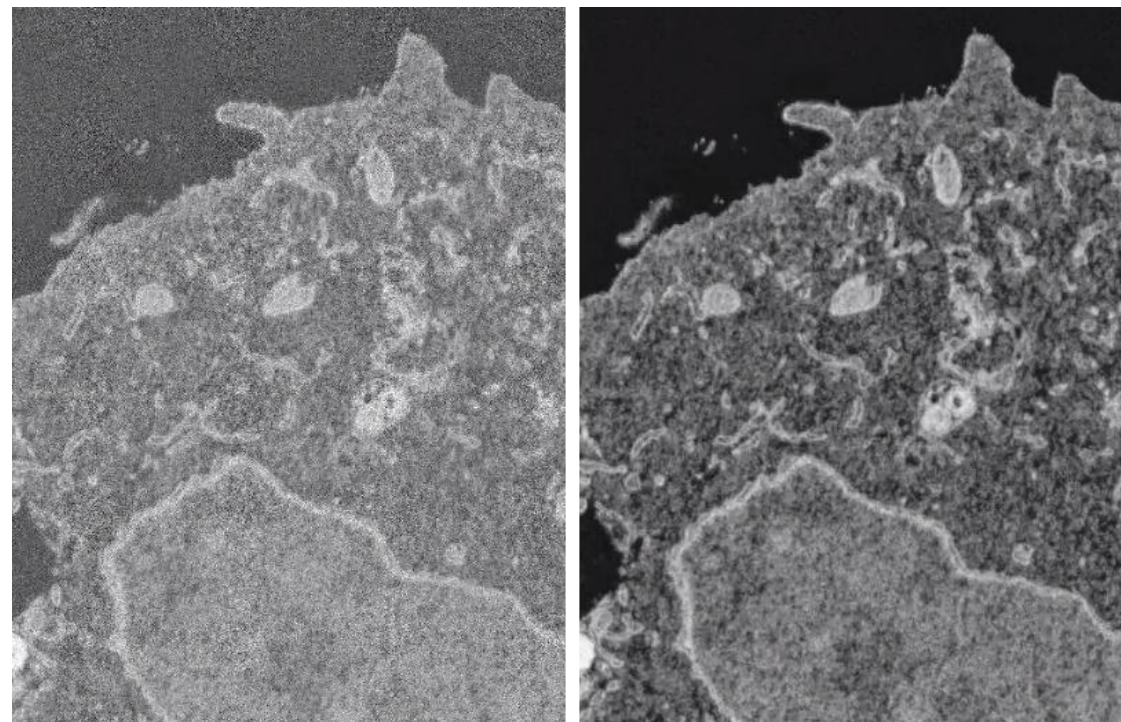
- Possibile l'addestramento di una rete neurale senza immagini pulite di riferimento (ground truth)

VS denoiser tradizionale

- Viene addestrato usando la stessa immagine sia come input sia come output
- Per cui la rete impara solo a copiare l'immagine (riproduce il rumore)

Noise2void utilizza una rete con «**blind spot**»

Il valore del pixel che deve essere predetto non è accessibile alla rete stessa durante la previsione



U-Net

U-Net funziona bene con Noise2Void perché la sua architettura è perfettamente adatta al modo in cui N2V impara il rumore.

N2V infatti nasconde o maschera un pixel chiede alla rete di **predirlo usando solo il contesto circostante**

quindi la rete deve capire **strutture locali e globali**

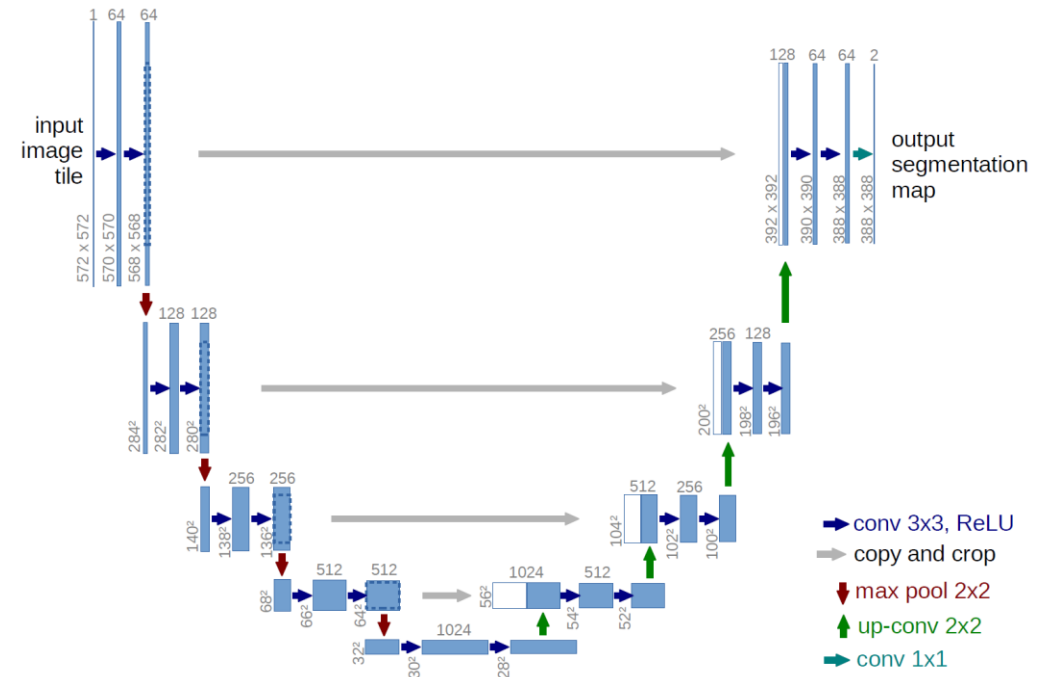
U-Net grazie ai 2 procedimenti di

- **encoder** → cattura il contesto ampio (forme, strutture)

- **decoder** → ricostruisce il dettaglio


skip connections → preservano informazione locale

In questo il pixel mascherato **non può essere copiato** deve essere **inferito dal contesto**



Implementazione del codice: U-Net

Con che parametri il modello verrà addestrato



```
{'algorithm_config': {'algorithm': 'n2v',  
    'loss': 'n2v',  
    'lr_scheduler': {'name': 'ReduceLROnPlateau',  
        'parameters': {}},  
    'model': {'architecture': 'UNet',  
        'conv_dims': 2,  
        'depth': 2,  
        'final_activation': 'None',  
        'in_channels': 1,  
        'independent_channels': True,  
        'n2v2': False,  
        'num_channels_init': 32,  
        'num_classes': 1},  
    'optimizer': {'name': 'Adam',  
        'parameters': {'lr': 0.0001}}},
```

Implementazione del codice – Training

L'addestramento del modello viene effettuato creando un oggetto CAREamist, inizializzato a partire dall'oggetto di configurazione definito in precedenza.

```
# Inizializza il modello CAREamist con la configurazione definita
careamist = CAREamist(source=config, work_dir="notebooks/models/bsd68")

print("Training starting now...")
careamist.train(train_source=train_path, val_source=val_path)
print("Training ended!") # Addestramento completato
```


Implementazione del codice – Predictions

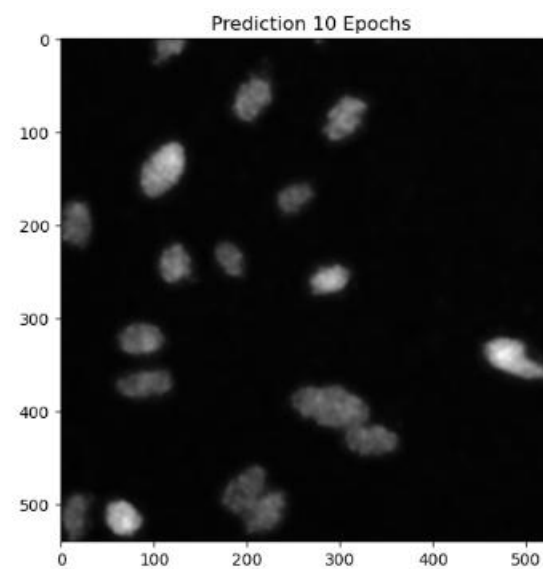
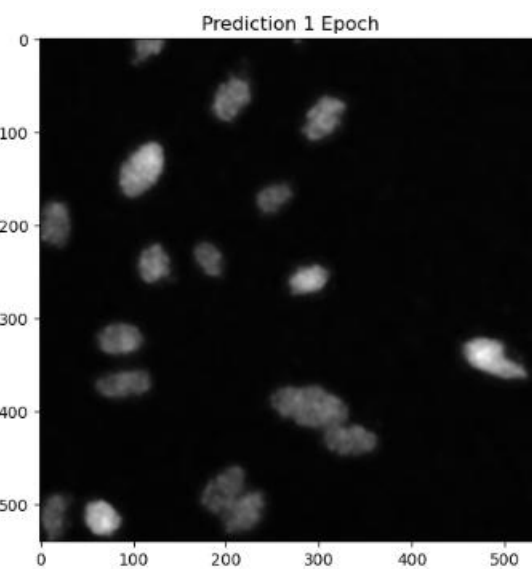
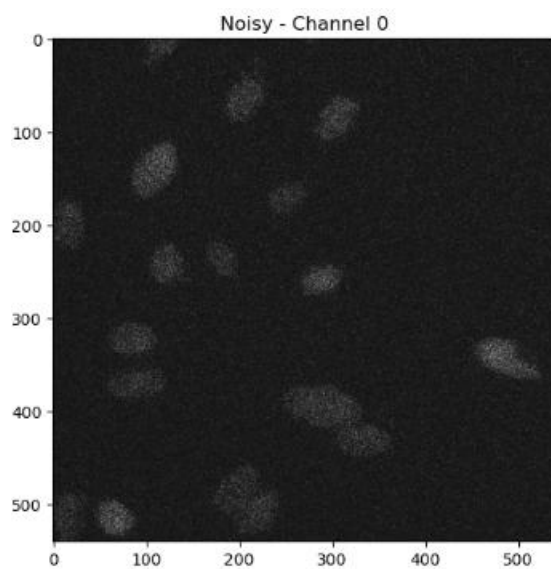
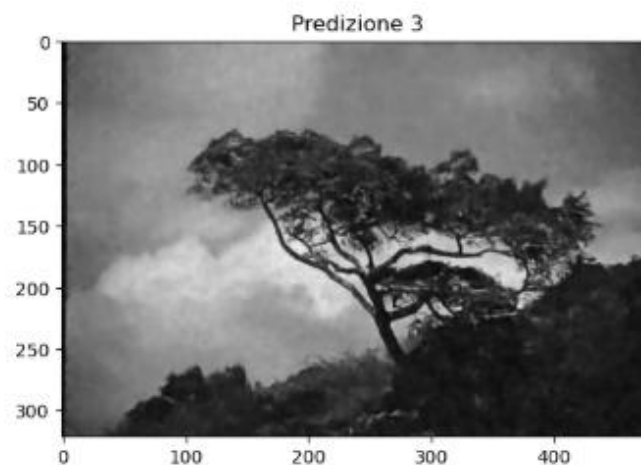
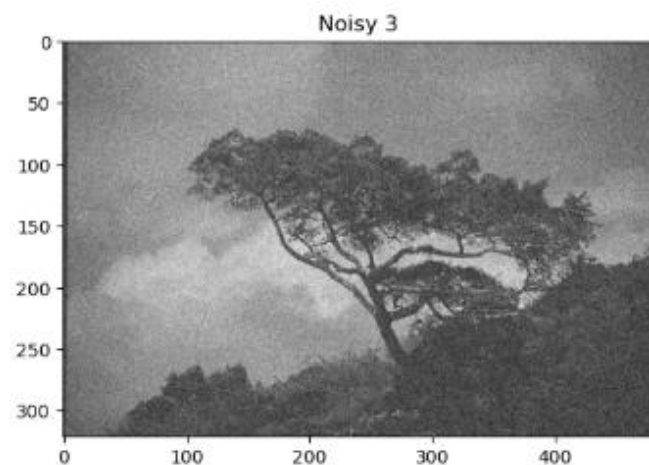
Il metodo `.predict` sfrutta l'**ultimo checkpoint salvato** durante l'addestramento per produrre le immagini denoised.

```
output_path = "notebooks/predictions/bsd68/predictions.tiff"

# Predizione delle immagini di test
prediction = careamist.predict(
    source=test_path,
    axes="YX",
    tile_size=(128, 128),
    tile_overlap=(48, 48),
    batch_size=1,
)
```

Questa fase consente di valutare il comportamento del modello su dati non visti durante l'addestramento e di confrontare le immagini predette con le corrispondenti **Ground Truth**, fornendo una misura oggettiva delle prestazioni del denoising

Risultati



Conclusioni

BSD68

- Nell'immagine **noisy** il rumore gaussiano è diffuso su tutta la scena
- La **predizione** mostra una forte riduzione del rumore
- Le strutture principali e i bordi restano ben definiti
- La struttura globale è simile al ground truth

Si osserva il tipico compromesso tra rimozione del rumore e preservazione dei dettagli

Jump_cells

- Nell'immagine **noisy** il rumore è molto elevato e le strutture cellulari sono poco visibili
- Dopo **1 epoch** si osserva già una riduzione del rumore e le cellule diventano distinguibili ma risulta ancora grezza e non completamente pulita
- Dopo **100 epoch** il rumore è ulteriormente ridotto
- Le forme cellulari appaiono ben definite su uno sfondo pulito
- Il modello apprende progressivamente la distinzione tra segnale e rumore
- Il denoising risulta efficace anche senza ground truth, senza introdurre artefatti

Riferimenti

1. **Ronneberger, O., Fischer, P., & Brox, T. (2015).** *U-Net: Convolutional Networks for Biomedical Image Segmentation*.^{*} MICCAI 2015.
<https://arxiv.org/abs/1505.04597>
2. **Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O. (2016).** *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*. MICCAI 2016.
3. **Mathis, A., et al. (2018)** *Deep Learning for Image Denoising: U-Net Architectures and Variants*.^{*} <https://www.mathstoml.com/u-net>
4. **Wikipedia – U-Net.** <https://en.wikipedia.org/wiki/U-Net>
5. **Ronneberger, O. (2015).** *U-Net architecture with skip connections*
https://www.researchgate.net/figure/U-Net-with-skip-connections_fig3_346417821
6. **Krull, A., Buchholz, T.O., & Jug, F. (2019).**
Noise2Void – Learning Denoising from Single Noisy Images.
CVPR 2019. <https://arxiv.org/abs/1811.10980>
7. **Ronneberger, O., Fischer, P., & Brox, T. (2015).** **U-Net: Convolutional Networks for Biomedical Image Segmentation**. **MICCAI 2015.**
<https://arxiv.org/abs/1505.04597>
8. <https://ai4life-mdc25.grand-challenge.org/>
9. Careamics Github - Noise2Void