Deep Learning Image Denoising

Federico Calandra | Ivan Pelizon

Indice

- Il problema del rumore
- I metodi classici
- Le metriche PSNR e SSIM
- MNIST e CIFAR10
- Strumenti utilizzati
- Struttura del codice
- Autoencoder per il Denoising
- Variational Autoencoder per il Denoising
- SOTA: Restormer
- Test su Restormer con dataset RENOIR
- Risultati

Il problema del rumore

Definizione

Sia $I \in \mathbb{R}^{m \times n}$ l'immagine originale con $x_{i,j} \in \mathbb{R}$ un pixel dell'immagine I. Sia $Y \in \mathbb{R}^{m \times n}$ l'immagine affetta da rumore e $y_{i,j} \in \mathbb{R}$ un suo pixel, allora

$$y_{i,j} = x_{i,j} + r_{i,j}$$

Dove $r_{i,j}$ con $i = 0, 1, ..., n$ e $j = 0, 1, ..., n$ sono v.a. indipendenti e identicamente distribuite (IID)

Le sorgenti del rumore

- Sensore (bassa luminosità, difetti)
- Rumore Elettronico (lettura dell'immagine dal sensore, amplificazione)
- Trasmissione (quantizzazione, compressione dell'immagine)
- Altri fattori (interferenza elettromagnetica, elaborazione digitale, condizioni atmosferiche, ecc.)

Gaussian Noise

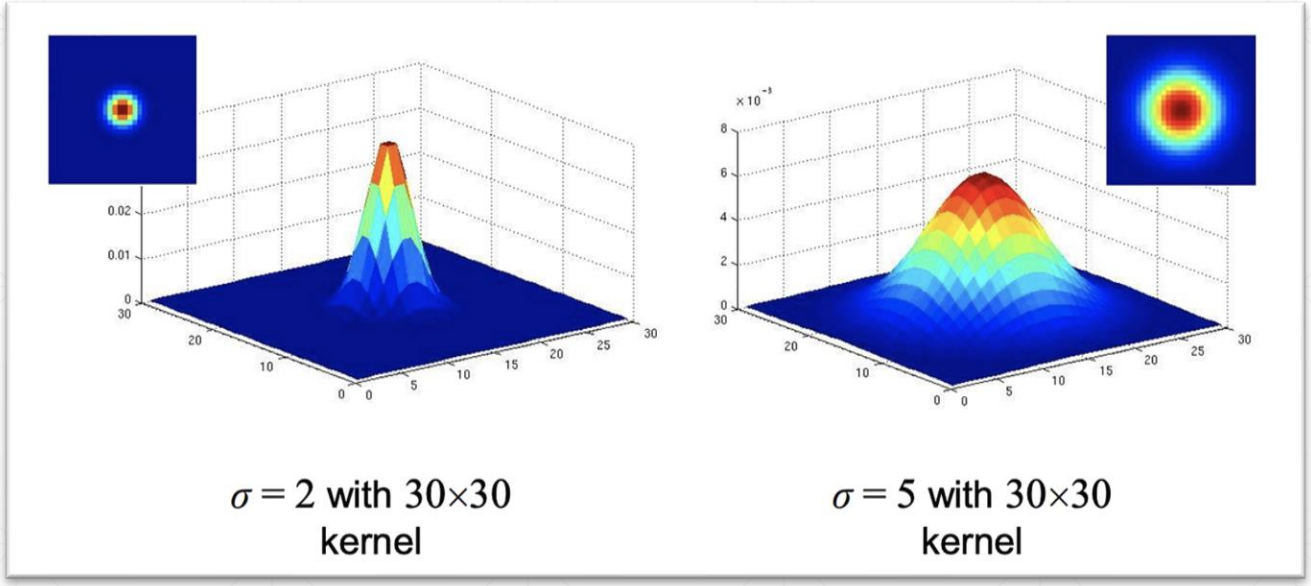
Salt and Pepper Noise

Altri tipi di rumore

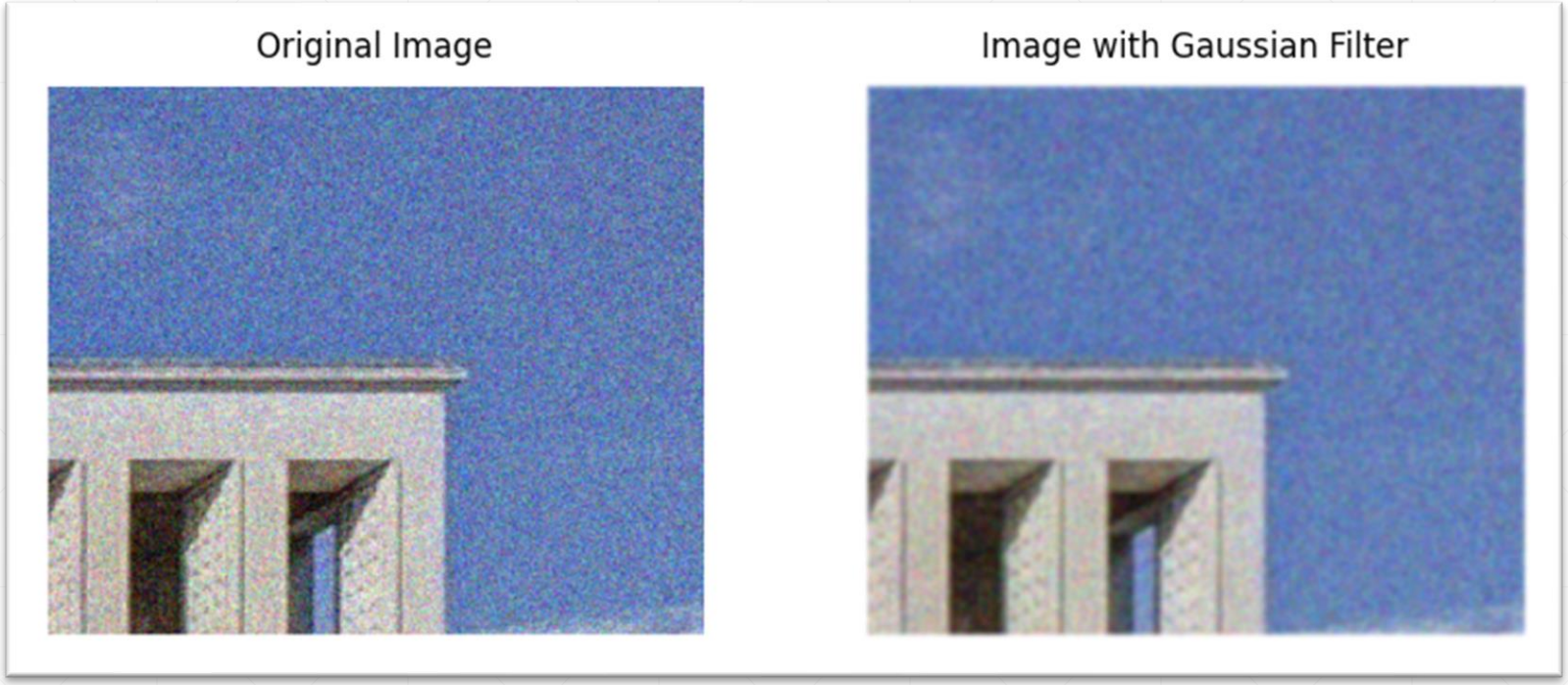
- Shot Noise
- Quantization Noise
- Color Noise

Metodi classici

Filtro Gaussiano

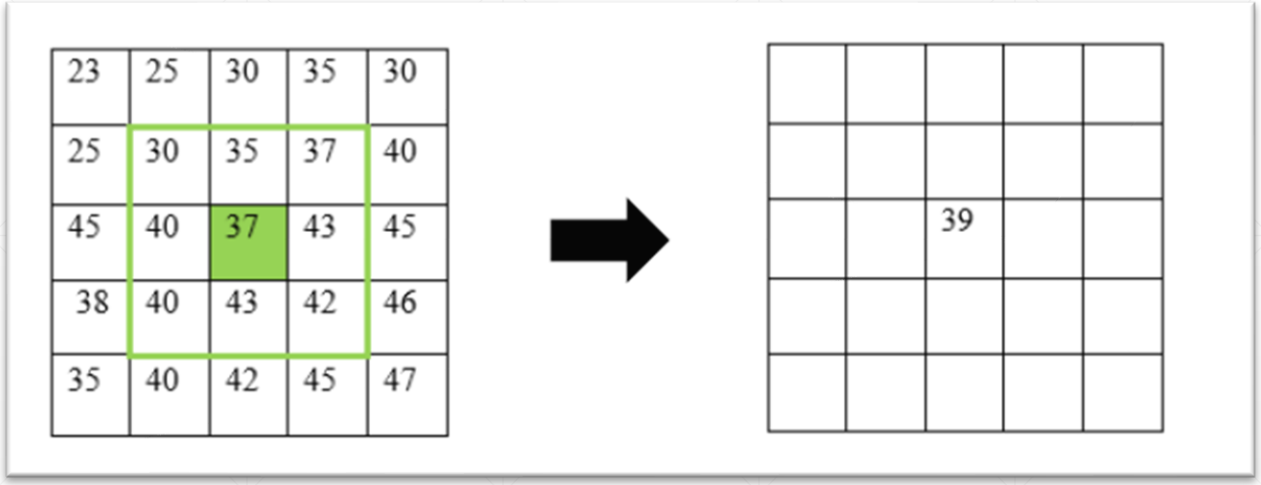


Filtro Gaussiano (ii)

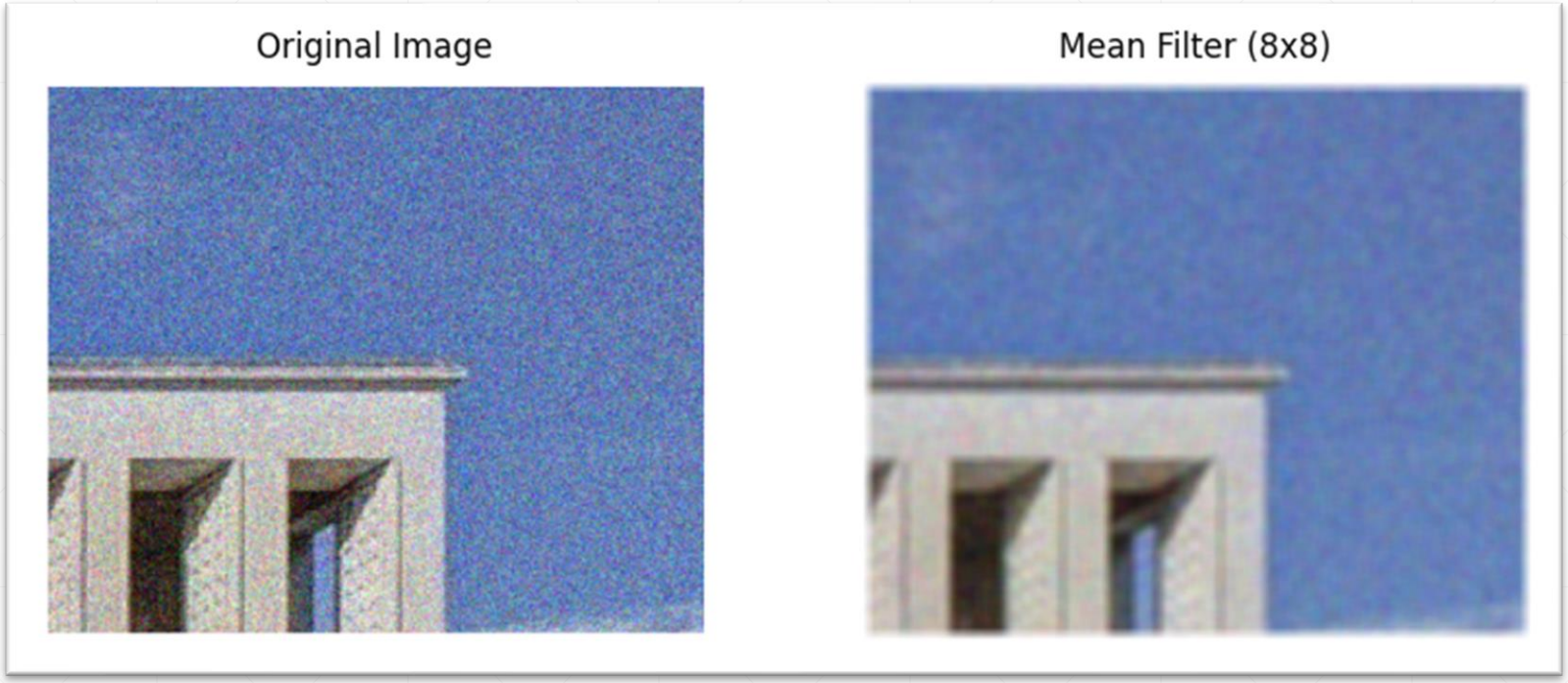


(sigma = 1.5)

Filtro Medio



Filtro Medio (ii)



Filtro Bilaterale

Sia $I \in \mathbb{R}^{m \times n}$ un'immagine affetta da rumore.

$$I_{\mathrm{f}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

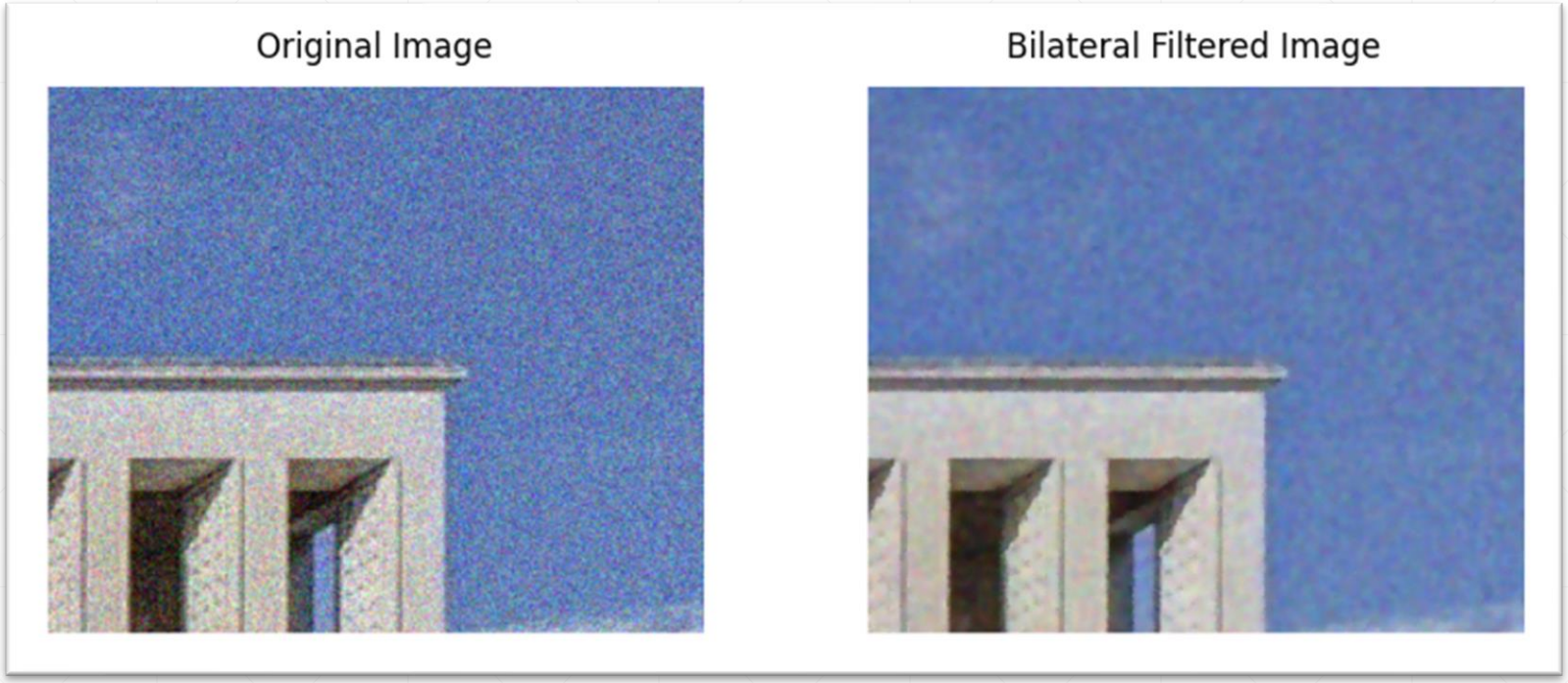
f_r è il range kernel (smorza le variazioni di intensità)
g_s è lo spatial kernel (smorza le differenze nelle coordinate)

W_p (termine di normalizzazione) è definito come:
$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$

Filtro Bilaterale (ii)

```
'''
Applicazione filtro bilaterale (funzione gaussiana per range
e spatial kernels)
'''
diameter = 9  # Dimensione di Omega (finestra di pixels)
sigma_color = 75  # sigma del range kernel (intensità)
sigma_space = 75  # sigma del spatial kernel (coordinate)
filtered_image = apply_bilateral_filter(input_image, diameter,
                                        sigma_color, sigma_space)
```

Filtro Bilaterale (iii)



Altri filtri

- Non-Local Means (NLM)
- Wiener Filter
- Anisotropic Diffusion

Is Denoising Dead?

Priyam Chatterjee, *Student Member, IEEE*, and Peyman Milanfar, *Fellow, IEEE*

***Abstract*—Image denoising has been a well studied problem in the field of image processing. Yet researchers continue to focus attention on it to better the current state-of-the-art. Recently proposed methods take different approaches to the problem and yet their denoising performances are comparable. A pertinent question then to ask is whether there is a theoretical limit to denoising performance and, more importantly, are we there yet? As camera manufacturers continue to pack increasing numbers of pixels per unit area, an increase in noise sensitivity manifests itself in the form of a noisier image. We study the performance bounds for the image denoising problem. Our work in this paper estimates a lower bound on the mean squared error of the denoised result and compares the performance of current state-of-the-art denoising methods with this bound. We show that despite the phenomenal recent progress in the quality of denoising algorithms, some room for improvement still remains for a wide class of general images, and at certain signal-to-noise levels. Therefore, image denoising is not dead—yet.**

erature on such performance limits exists for some of the more complex image processing problems such as image registration [7], [8] and super-resolution [9]–[12]. Performance limits to object or feature recovery in images in the presence of point-wise degradation has been studied by Treibitz *et al.* [13]. In their work, the authors study the effects of noise among other degradations and formulate expressions for the optimal filtering parameters that define the resolution limits to recovering any given feature in the image. While their study is practical, it does not define statistical performance limits to denoising general images. In [14], Voloshynovskiy *et al.* briefly analyze the performance of MAP estimators for the denoising problem. However, our bounds are developed in a much more general setting and, to the best of our knowledge, no comparable study currently exists for the problem of denoising. The present

...2010

Si iniziava a pensare che I metodi classici non potessero essere migliorati più di tanto...

Image Denoising: The Deep Learning Revolution and Beyond – A Survey Paper –

Michael Elad, Bahjat Kawar and Gregory Vaksman
The Computer Science Department, Technion – Israel Institute of Technology
email: {elad,bahjat.kawar,grishavak}@cs.technion.ac.il

Abstract.

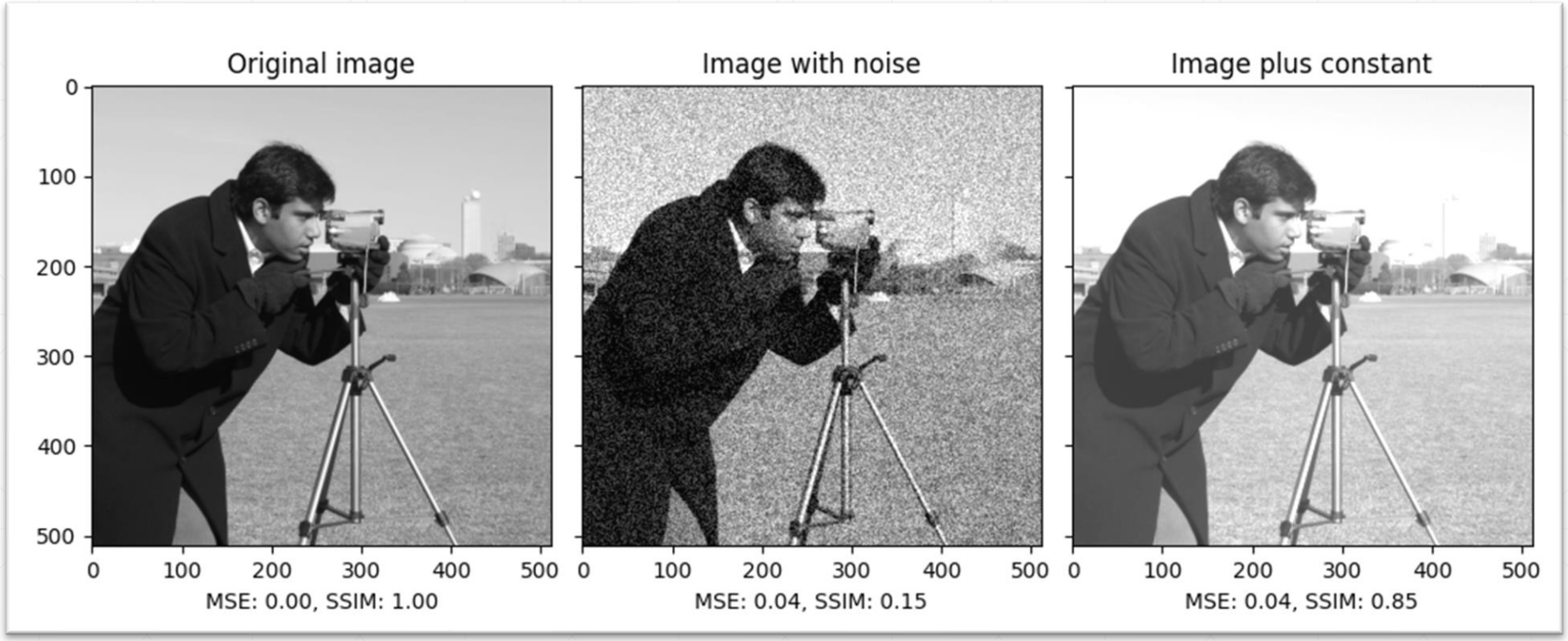
Image denoising – removal of additive white Gaussian noise from an image – is one of the oldest and most studied problems in image processing. An extensive work over several decades has led to thousands of papers on this subject, and to many well-performing algorithms for this task. Indeed, ten years ago, these achievements have led some researchers to suspect that "Denoising is Dead", in the sense that all that can be achieved in this domain has already been obtained. However, this turned out to be far from the truth, with the penetration of deep learning (DL) into the realm of image processing. The era of DL brought a revolution to image denoising, both by taking the lead in today's ability for noise suppression in images, and by broadening the scope of denoising problems being treated. Our paper starts by describing this evolution, highlighting in particular the tension and synergy that exist between classical approaches and modern Artificial Intelligence (AI) alternatives in design of image denoisers.

Deep Learning Revolution

- Ha inizio con l'introduzione dei modelli convoluzionali CNN (2012)
- Surclassano rapidamente i modelli classici
- Generalmente basati su una struttura encoder-decoder

Metriche PSNR e SSIM

Perché non MSE?



Peak Signal-to-Noise Ratio (PSNR)

- Rapporto tra la massima potenza del segnale e la potenza del rumore che lo affligge

Sia $I \in \mathbb{R}^{m \times n}$ l'immagine originale e sia $K \in \mathbb{R}^{m \times n}$ l'immagine affetta da rumore, allora

$$MSE = \frac{1}{m\,n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) && (1) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) && (2) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) && (3) \end{aligned}$$

MAX_I è il massimo valore assunto dai pixel dell'immagine

Structural Similarity Index Measure (SSIM)

- Obiettivo: creare un modello di valutazione della qualità delle immagini basato sulla percezione visiva umana
- Tale percezione si basa in larga parte sull'estrazione di informazioni strutturali dalla scena visiva
- Informazione strutturale: i pixel hanno una forte inter-dipendenza tra loro (specialmente se spazialmente vicini)
- Differenza con MSE e PSNR: si basano entrambi su una stima assoluta degli errori dei singoli pixels

Structural Similarity Index Measure (SSIM)

600 IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 4, APRIL 2004

Image Quality Assessment: From Error Visibility to Structural Similarity

Zhou Wang, *Member, IEEE*, Alan Conrad Bovik, *Fellow, IEEE*, Hamid Rahim Sheikh, *Student Member, IEEE*, and Eero P. Simoncelli, *Senior Member, IEEE*

SSIM si basa su 3 componenti:

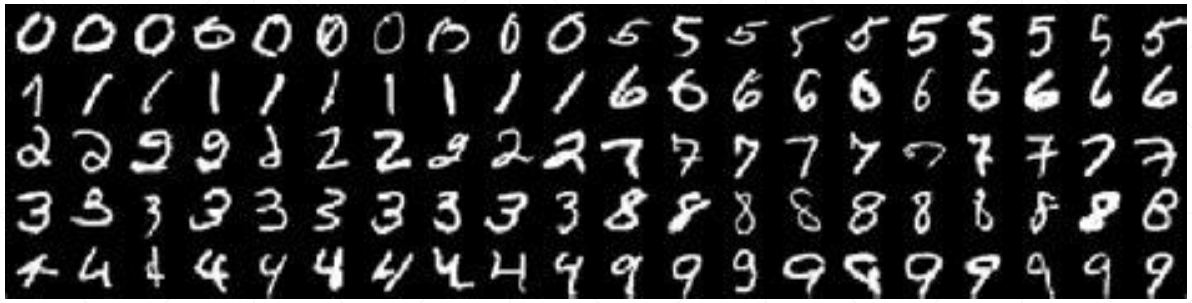
- Luminosità
- Contrasto
- Struttura

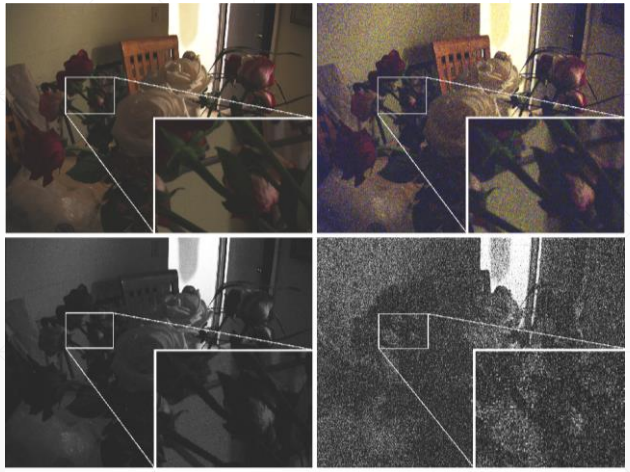
Assume valori compresi tra [-1, 1]

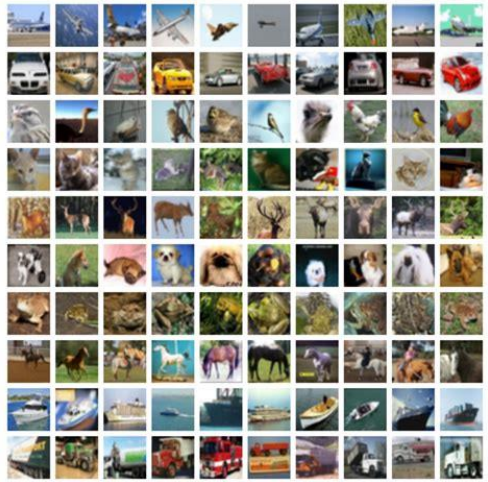
Strumenti utilizzati

Dataset considerati

- MNIST
  - 60000 training, 10000 testing
  - 28x28x1
- CIFAR10
  - 50000 training, 10000 testing
  - 10 categorie
  - 32x32x3
- Renoir (solo per Restormer)
  - Immagini con rumore reale
  - 3000x3000x3







Strumenti utilizzati

- Tensorflow, Keras
  - Keras layers API
  - Keras Models API
  - Losses, optimizers
- GitHub
  - Lavoro di gruppo
  - CI
- Google Colab
  - Jupyter Notebooks con GPU remota dedicata

SCAN ME

README.md

Deep Learning Project

build and test CPU passing | build and test GPU passing | Build LaTeX report passing | license GPL-3.0

Exam project for the Deep Learning course @ UniTS

Give it a try!

The quickest way to get started is to train the models using the provided Jupyter Notebooks. You can try them out quickly on Google Colab:

- Denoising Autoencoder trained on MNIST Open in Colab
- Denoising Autoencoder trained on CIFAR10 Open in Colab
- Variational Autoencoder trained on MNIST Open in Colab
- Variational Autoencoder trained on CIFAR10 Open in Colab
- Denoising Variational Autoencoder trained on MNIST Open in Colab
- Restormer Open in Colab

Otherwise, check out the setup section.

Project description

Image denoising is a fundamental problem in computer vision, with a wide range of real-world applications, including improving the quality of images for object segmentation, detection, tracking, and more. This project aims to explore and implement state-of-the-art deep learning models for image denoising, with a particular focus on Restormer, a Transformer based denoising network.

Deep Learning has demonstrated remarkable success in this field [1], and our project would leverage this technology to address the issue of noisy images. We'd delve into the principles of deep learning, especially Convolutional Neural Networks (CNNs), and understand how these networks can be trained to remove noise and enhance image quality.

One of the key highlights of this project is the exploration of the Restormer architecture, a novel approach to image denoising[2]. The Restormer model efficiently combines the strengths of Transformers and addresses the limitations of CNNs, making it particularly suitable for high-resolution image restoration tasks. By investigating the Restormer

Pagina iniziale del repository

Struttura del repository

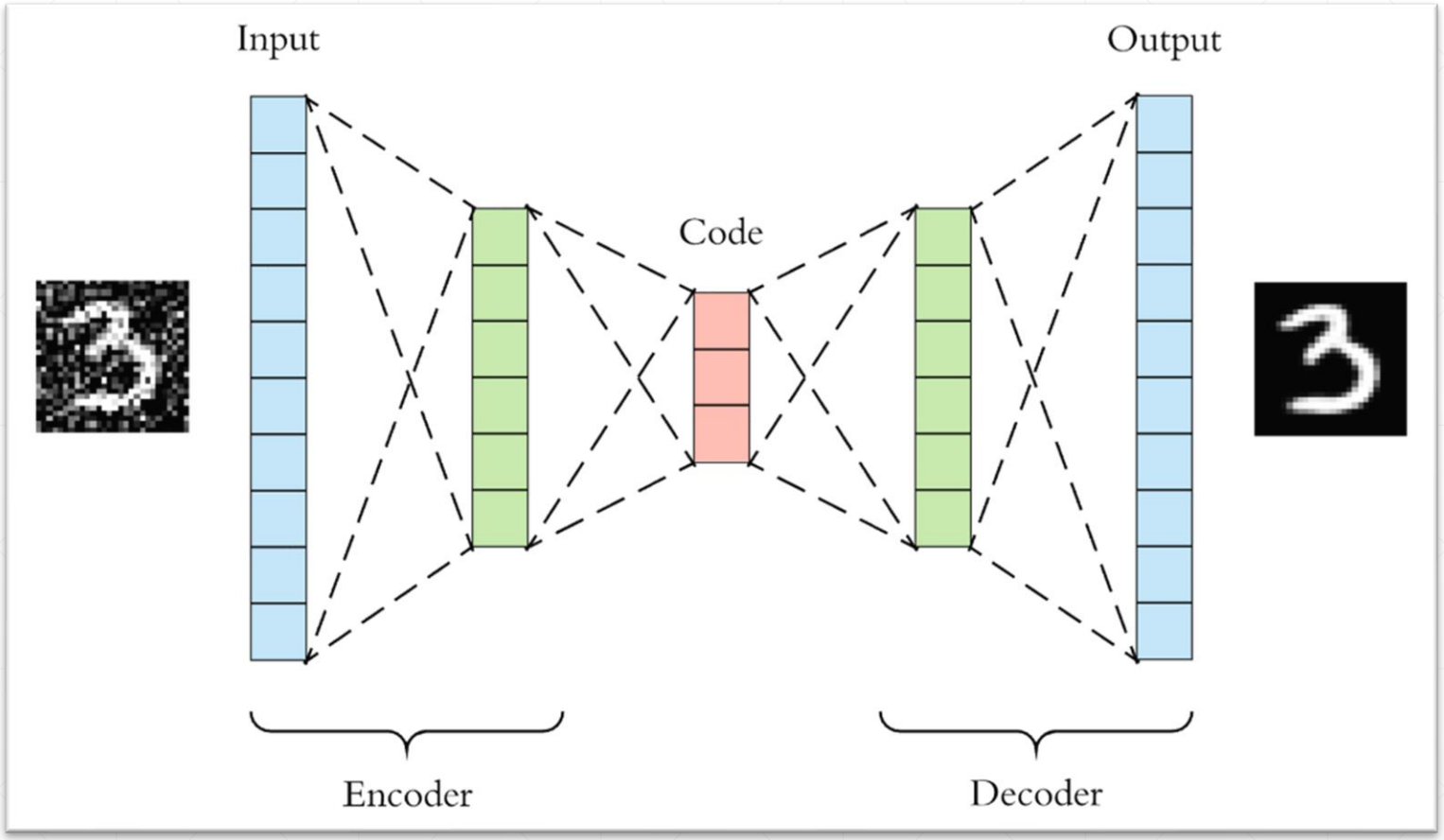
- Classi dataset
  - Aggiunta rumore opzionale, inizializzazione dataset
- Classi modelli
  - Definizione strati
  - Estrazione encoder e decoder
- Visualizzazione risultati

peiva-git Merge pull request #84 from peiva-git/81-metriche-renoir-trad ... 9d49631 15 minutes ago 177 commits

.github	Added issue templates	3 weeks ago
.idea	Added preprocessing command line arguments	2 days ago
conda	Updated notebook and GPU conda env	last week
dlproject	Refactor	2 hours ago
images	Added images and model weights	4 days ago
models	Added images and model weights	4 days ago
report	Update main.tex	2 weeks ago
results	Added command line arguments, added computed metrics for tradition...	2 hours ago
tests	Setup project template	3 weeks ago
tools	Refactor	2 hours ago
.gitattributes	Added conf file, see all the necessary steps at the provided link	last week
LICENSE	Initial commit	2 months ago
ProjectProposal_CalandraPelizon.pdf	Added corrected Project Proposal	3 weeks ago
README.md	Bug fix	2 days ago
Restormer.ipynb	Added Restormer notebook	last week
denoising_simple_vae_mnist.ipynb	Bug corrected, need to reshape noisy data as well	4 days ago
requirements.txt	Updated dependencies to match Colab environment. Added results file	2 hours ago
restormer_evaluation.ipynb	output removed	5 days ago
setup.py	Setup project template	3 weeks ago
simple_ae_cifar10.ipynb	Added preprocessing command line arguments	2 days ago
simple_ae_mnist.ipynb	Added preprocessing command line arguments	2 days ago
simple_model.ipynb	Bug fix	5 days ago
simple_vae_cifar10.ipynb	Update simple_vae_cifar10.ipynb	5 days ago
simple_vae_mnist.ipynb	Added preprocessing command line arguments	2 days ago

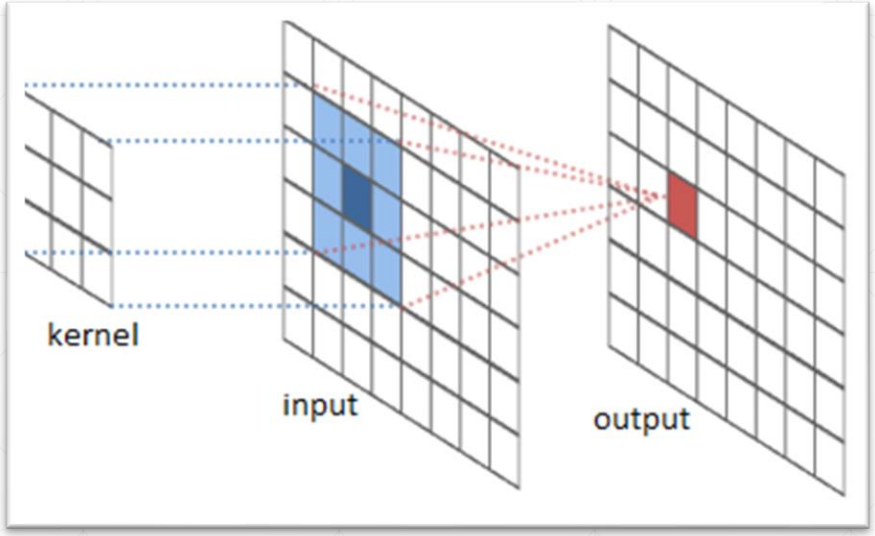
Autoencoder

Encoder-Decoder



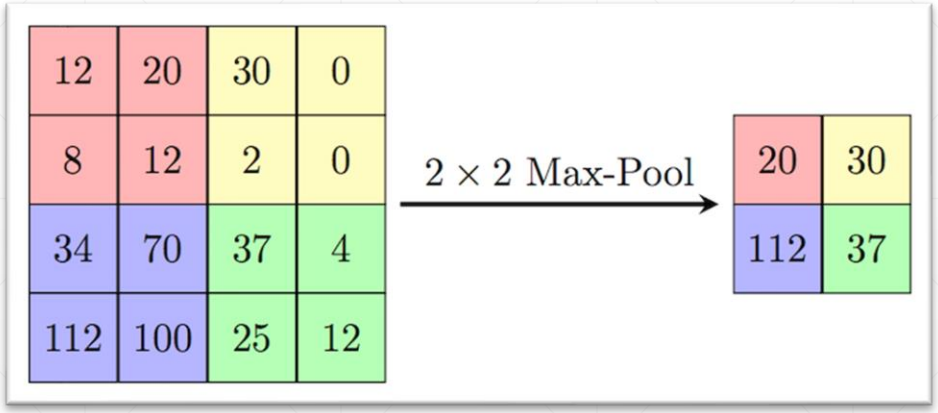
Encoder

- 2 layer convoluzionali con gli stessi parametri
- 32 filtri ciascuno
- Kernel 3x3
- ReLU
- Padding ('same')



Encoder (ii)

- Max Pooling dopo ogni layer convoluzionale



Decoder

- 2 transposed convolutional layers (upsampling)
- 32 filtri
- Kernel 3x3
- Stride = 2
- ReLU

Convoluzione trasposta

Input

0	1
2	3

Transposed Conv (stride 2)

Kernel

0	1
2	3

=

0	0		
0	0		

+

		0	1
		2	3

+

0	2		
4	6		

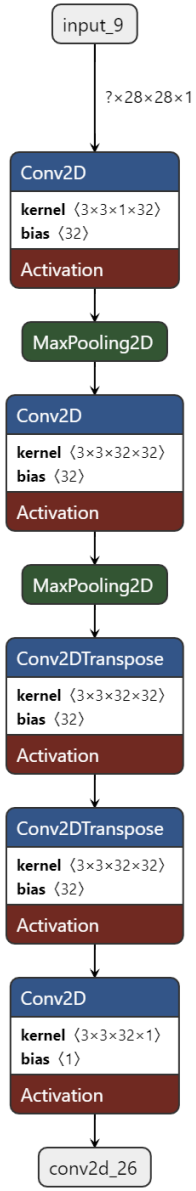
+

		0	3
		6	9

=

0	0	0	1
0	0	2	3
0	2	0	3
4	6	6	9

Output



Struttura della rete

L'ultimo layer convoluzionale ha un numero di filtri pari ai canali dell'immagine di input

Addestramento

- Svolto per vari livelli di rumore
- Funzione di Cross-Entropy loss
- Adam
  - $lr = 0{,}001$
  - $\beta_1 = 0{,}9,\ \beta_2 = 0{,}999$
- 100 epoche

Variational Auto-Encoder

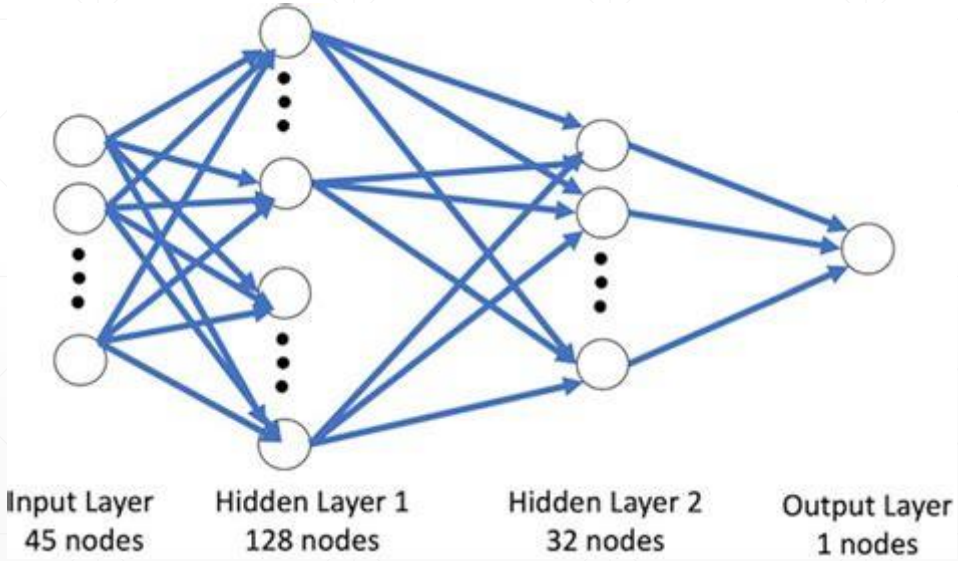
Preparazione dataset

- Normalizzazione immagini
- Ridimensionamento dati: un perceptron per pixel
- Considerate solo immagini bianco e nero (grayscale)
- Aggiunta rumore artificiale

Encoder

- Strato fully-connected intermedio con 64 perceptron
  - ReLU
- Latent space bidimensionale

Strato fully-connected

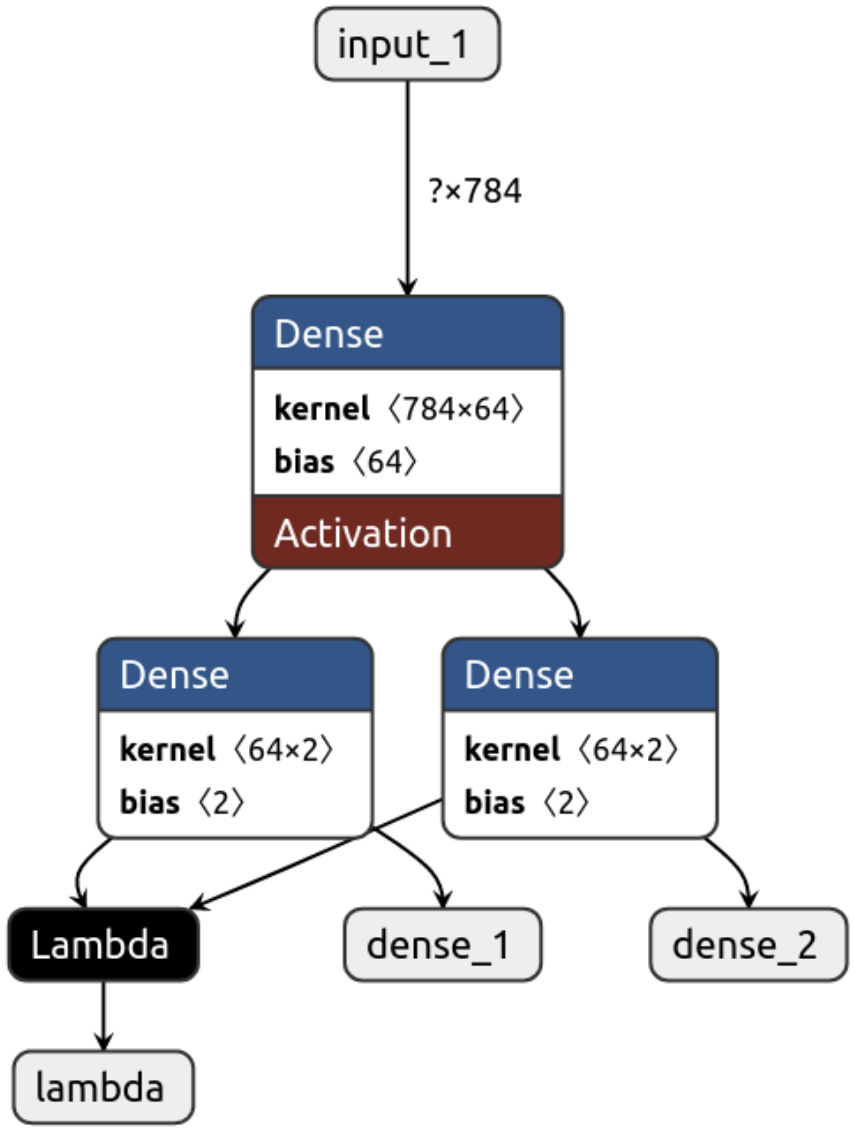


Latent space

- Rete addestrata per ricavare distribuzione di probabilità dei dati
  - Si assume distribuzione normale
- Codifica tramite parametri μ e σ
  - Codificati ciascuno da uno strato fully-connected
  - Activation lineare
- Generazione campioni tramite $z = \mu + e^{\sigma x}$
  - x campionato da distribuzione normale con media zero e deviazione standard 0,1

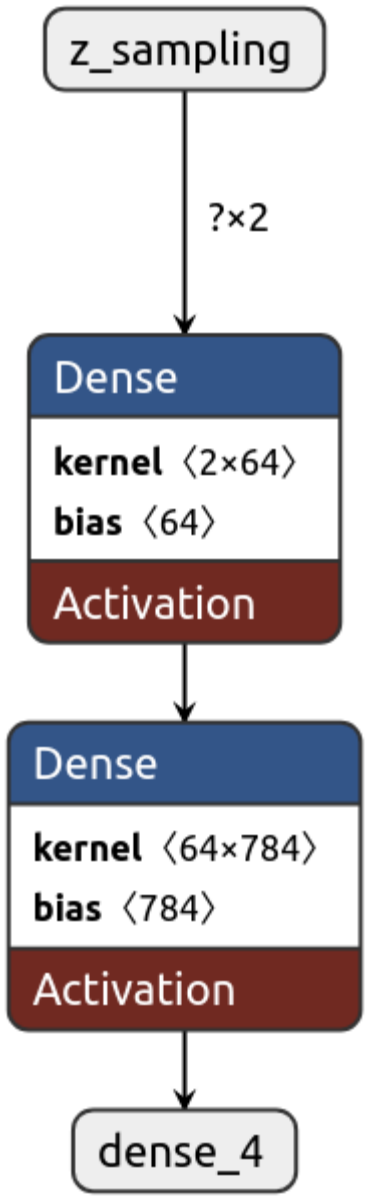
Decoder

- Strato fully-connected intermedio
  - ReLU
- Strato finale con stessa dimensione dell’input
  - Sigmoid



Struttura della rete

Encoder



Struttura della rete

Decoder

Addestramento

- Encoding e denoising
- Funzione di Cross-Entropy loss
- Ottimizzatore Adam
  - $lr = 0{,}001$
  - $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$
- 100 epoche

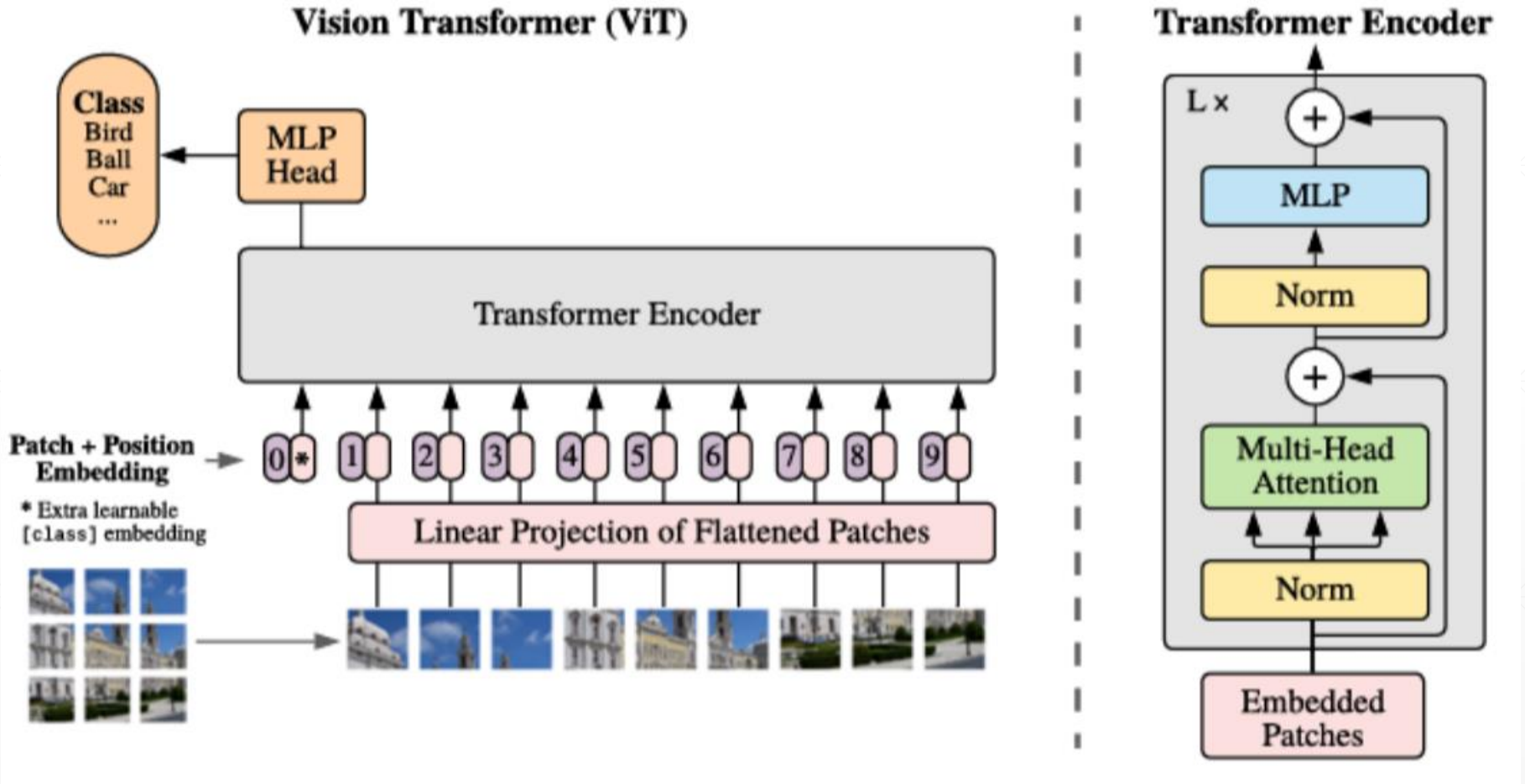
Restormer

Restoration Transformer

CNNs

- Hanno un *receptive field* limitato (difficoltà a modellare dipendenze tra pixel lontani)
- I filtri convoluzionali hanno pesi statici in fase di inferenza, i.e. non si adattano al contenuto dell'input (flessibilità)
- => self-attention

Vision Transformer - ViT



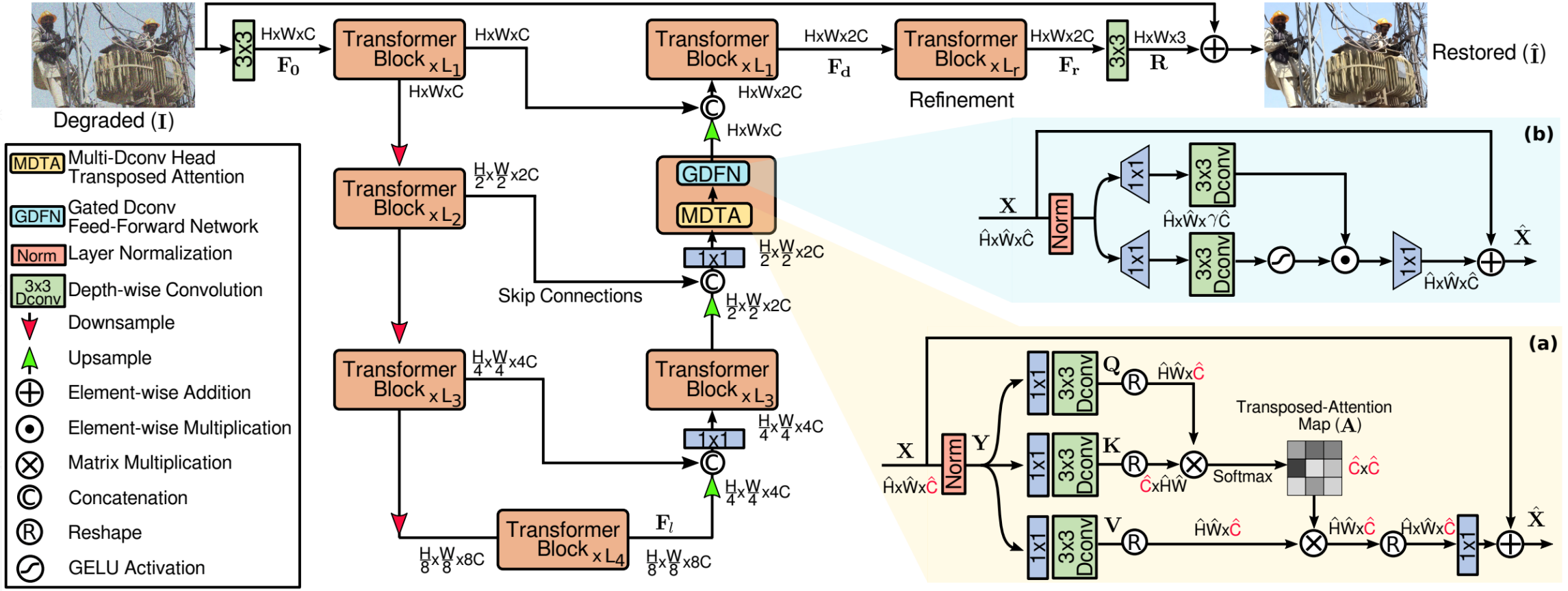
Immagini ad alta risoluzione

- La complessità computazionale cresce con il quadrato della risoluzione spaziale, i.e. O(NM^2)
- Non praticabile con immagini ad alta risoluzione
- Esempio: N = 6016, M = 4016, NxM = 2,4x10^7, (NxM)^2 = 5,8x10^14

Immagini ad alta risoluzione - Restormer

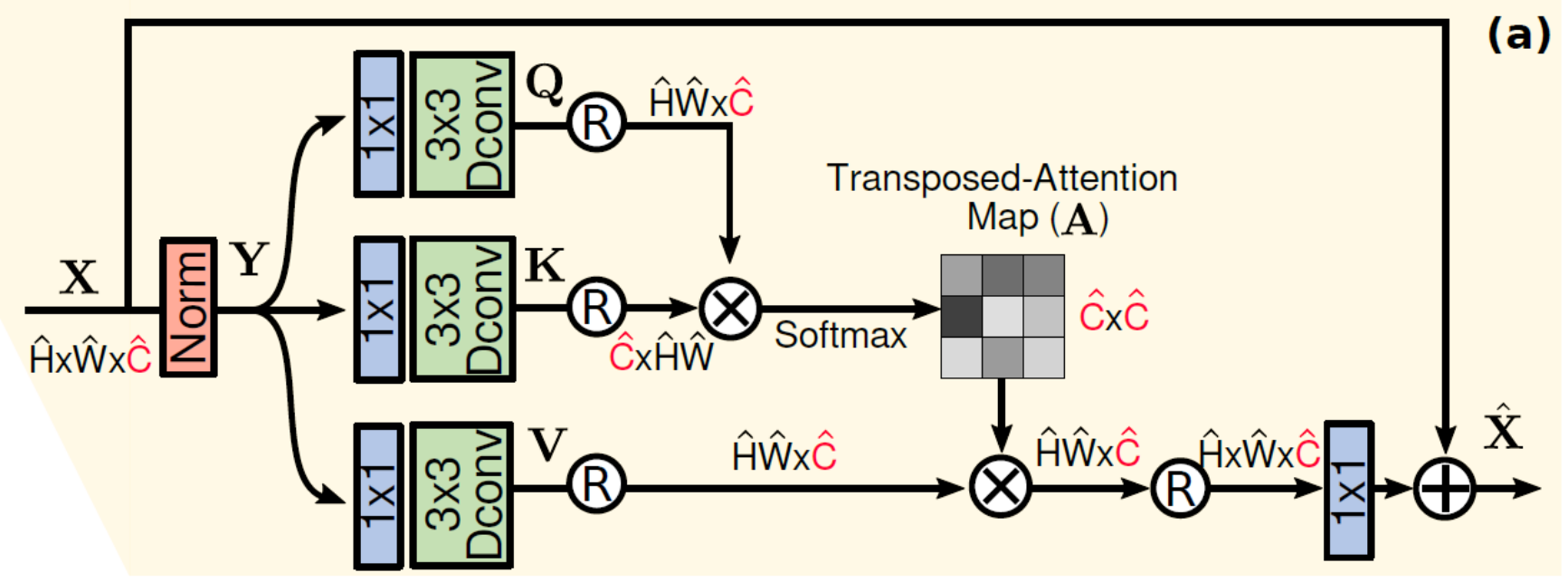
- Adatto anche per immagini ad alte risoluzioni (complessità lineare)
- Innovazioni introdotte dagli autori:
  - Multi-Dconv Head Transposed Attention (MDTA)
  - Gated-Dconv Feed-Forward Network (GDFN)
- Architettura encoder-decoder (Transformers)

Architettura Restormer



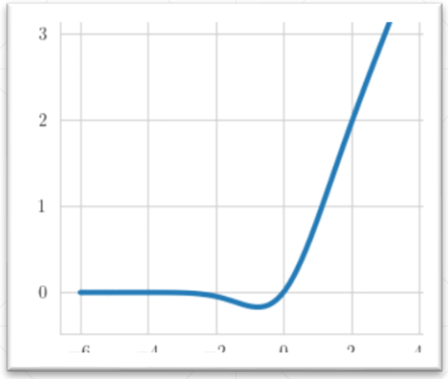
Multi-Dconv Head Transposed Attention

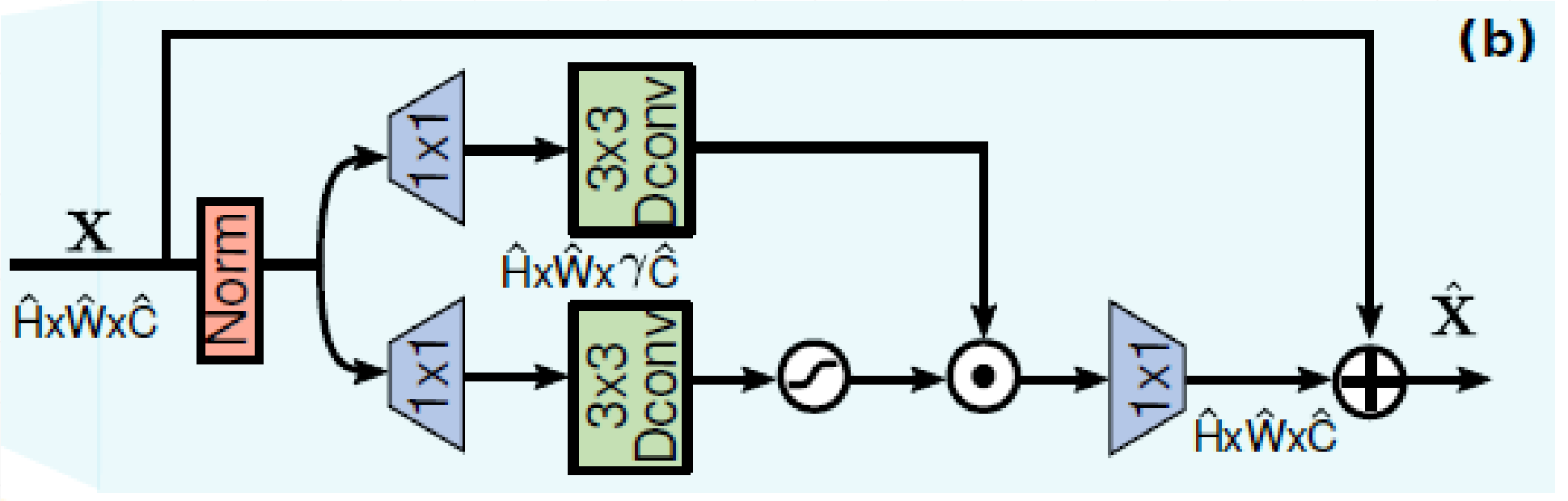
- Complessità lineare
- *Attention map* CxC
- Depth-Convolution (Dconv)
- Reshape di Q, K, V



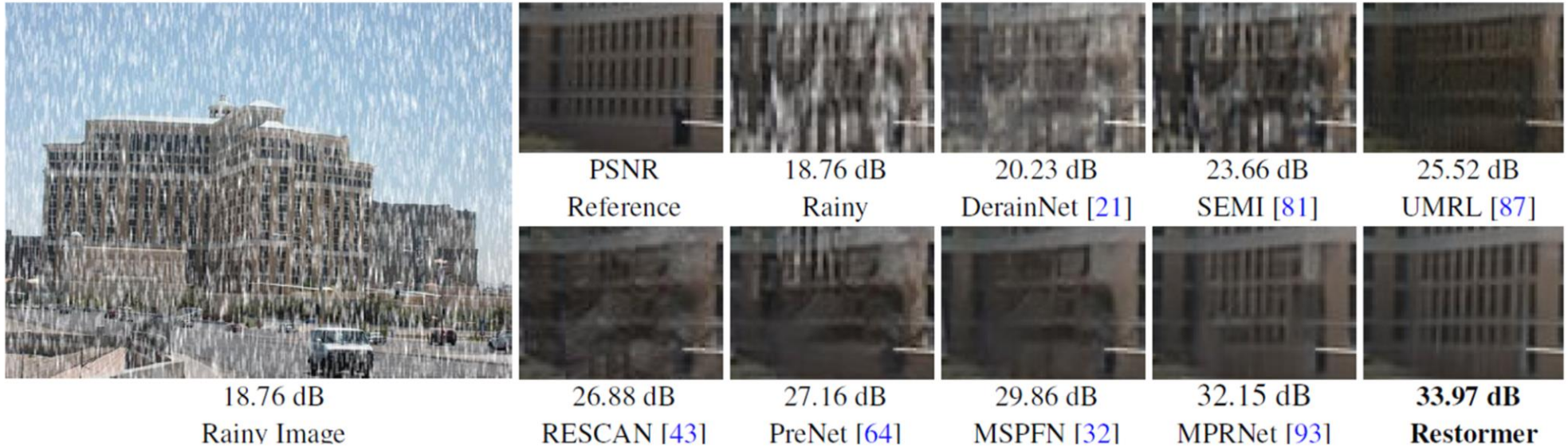
Gated-Dconv Feed-Forward Network

- Sostituisce MLP
- Meccanismo di *gating*
- GeLu (Gaussian Error Linear Unit

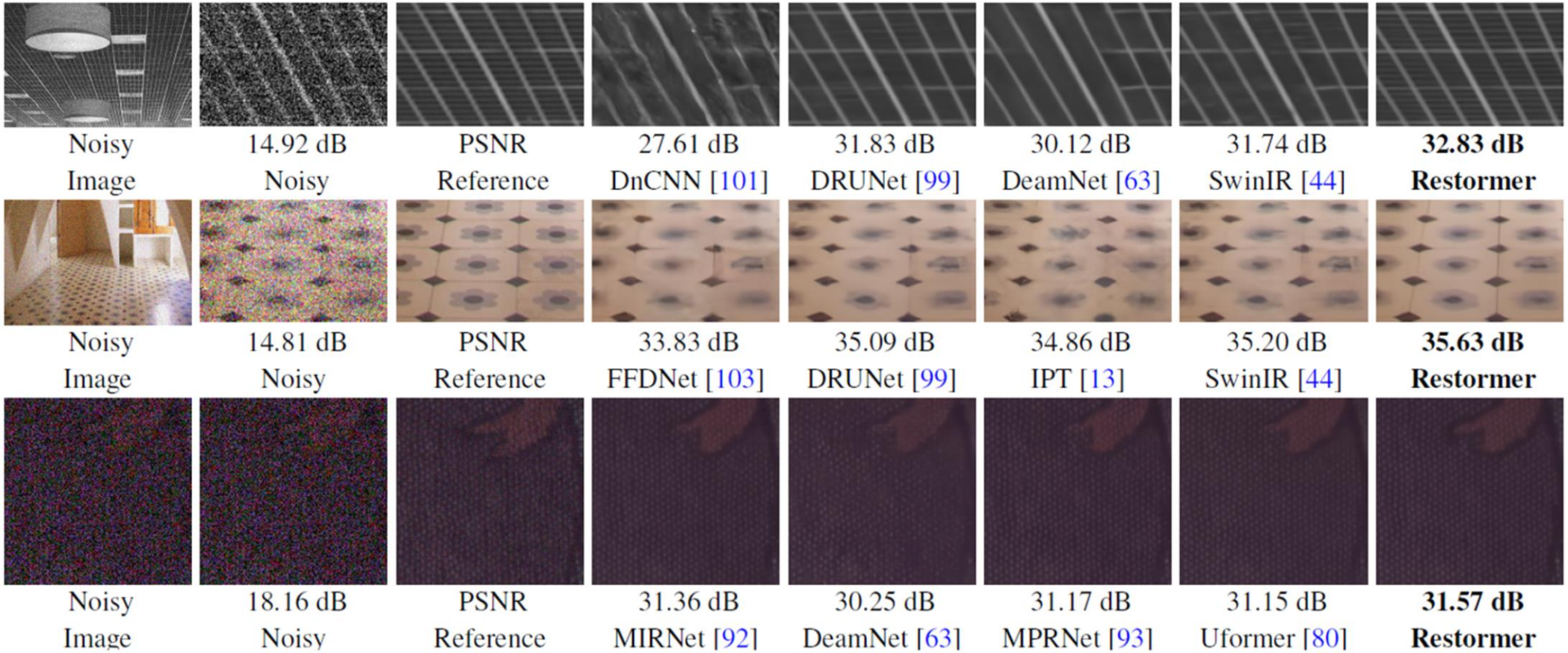




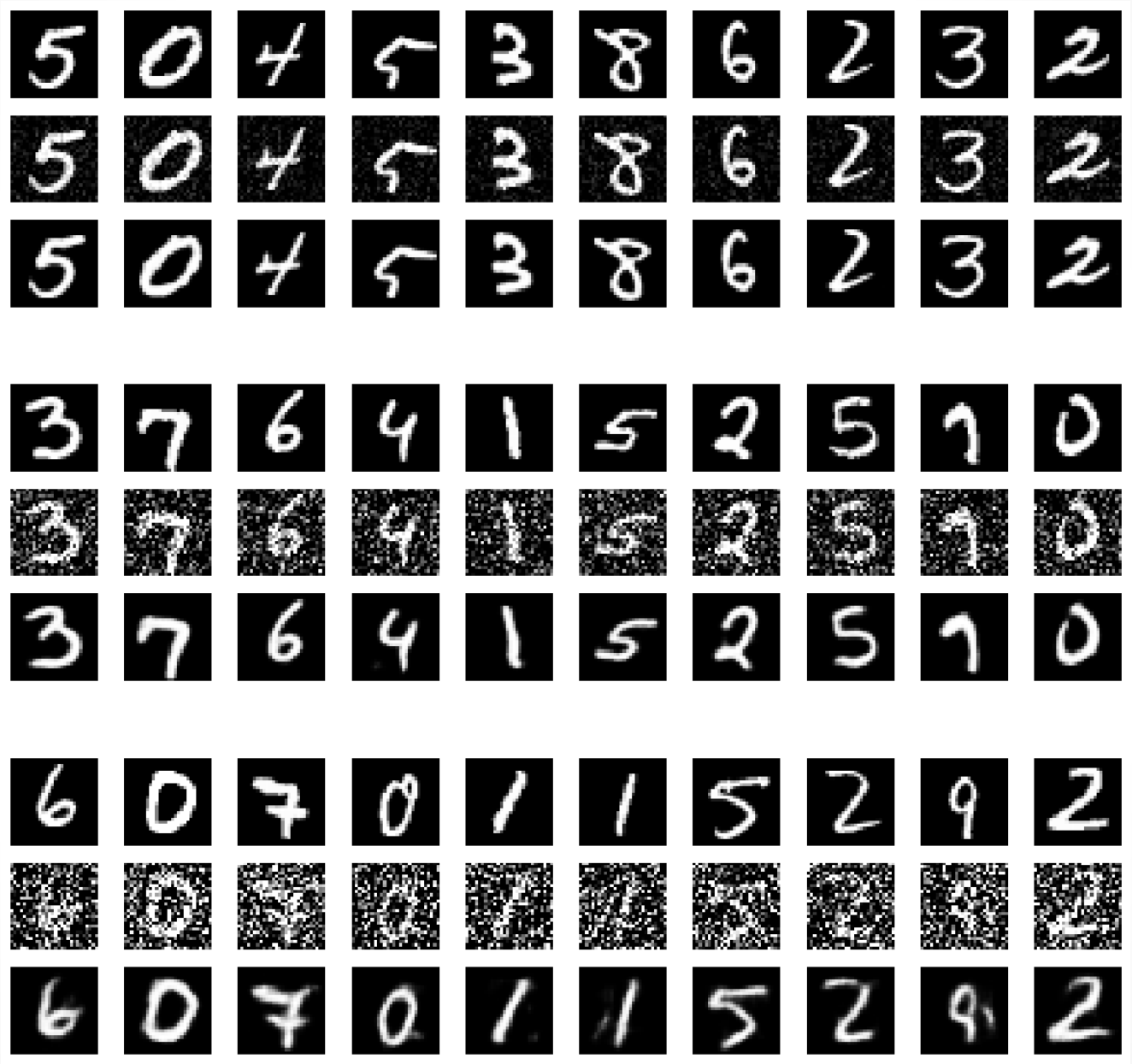
PSNR
Reference
19.45 dB
Blurry
23.85 dB
Gao et al. [23]
23.56 dB
DBGAN [100]
23.86 dB
MTRNN [58]
19.45 dB
Blurry Image
24.85 dB
DMPHN [94]
25.20 dB
Suin et al. [71]
25.67 dB
MPRNet [93]
24.33 dB
MIMO-UNet+ [14]
26.96 dB
Restormer


PSNR
Reference
18.76 dB
Rainy
20.23 dB
DerainNet [21]
23.66 dB
SEMI [81]
25.52 dB
UMRL [87]
18.76 dB
Rainy Image
26.88 dB
RESCAN [43]
27.16 dB
PreNet [64]
29.86 dB
MSPFN [32]
32.15 dB
MPRNet [93]
33.97 dB
Restormer

Dataset	Method	DnCNN [101]	BM3D [15]	CBDNet* [25]	RIDNet* [6]	AINDNet* [35]	VDN [89]	SADNet* [12]	DANet+* [90]	CycleISP* [91]	MIRNet [92]	DeamNet* [63]	MPRNet [93]	DAGL [55]	Uformer [80]	**Restormer** (Ours)
SIDD [1]	PSNR ↑	23.66	25.65	30.78	38.71	39.08	39.28	39.46	39.47	39.52	39.72	39.47	39.71	38.94	39.77	**40.02**
	SSIM ↑	0.583	0.685	0.801	0.951	0.954	0.956	0.957	0.957	0.957	0.959	0.957	0.958	0.953	0.959	**0.960**
DND [60]	PSNR ↑	32.43	34.51	38.06	39.26	39.37	39.38	39.59	39.58	39.56	39.88	39.63	39.80	39.77	39.96	**40.03**
	SSIM ↑	0.790	0.851	0.942	0.953	0.951	0.952	0.952	0.955	0.956	0.956	0.953	0.954	0.956	0.956	**0.956**

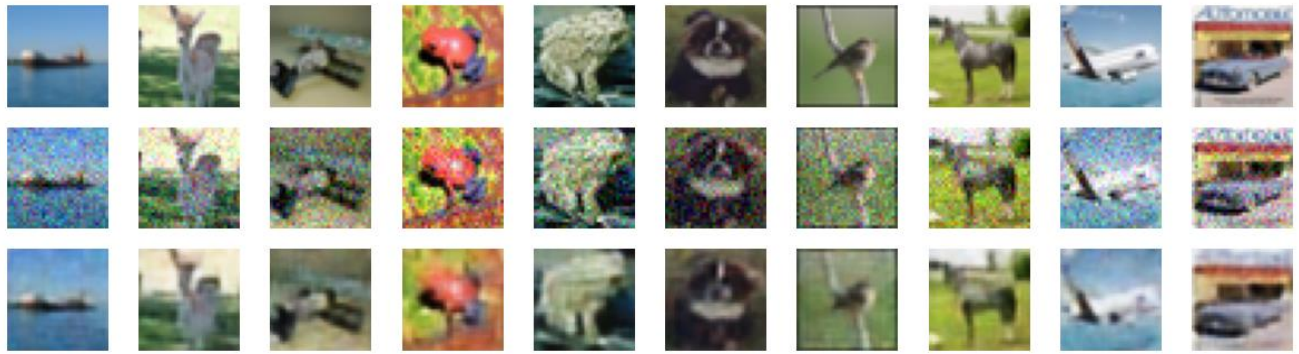


Risultati



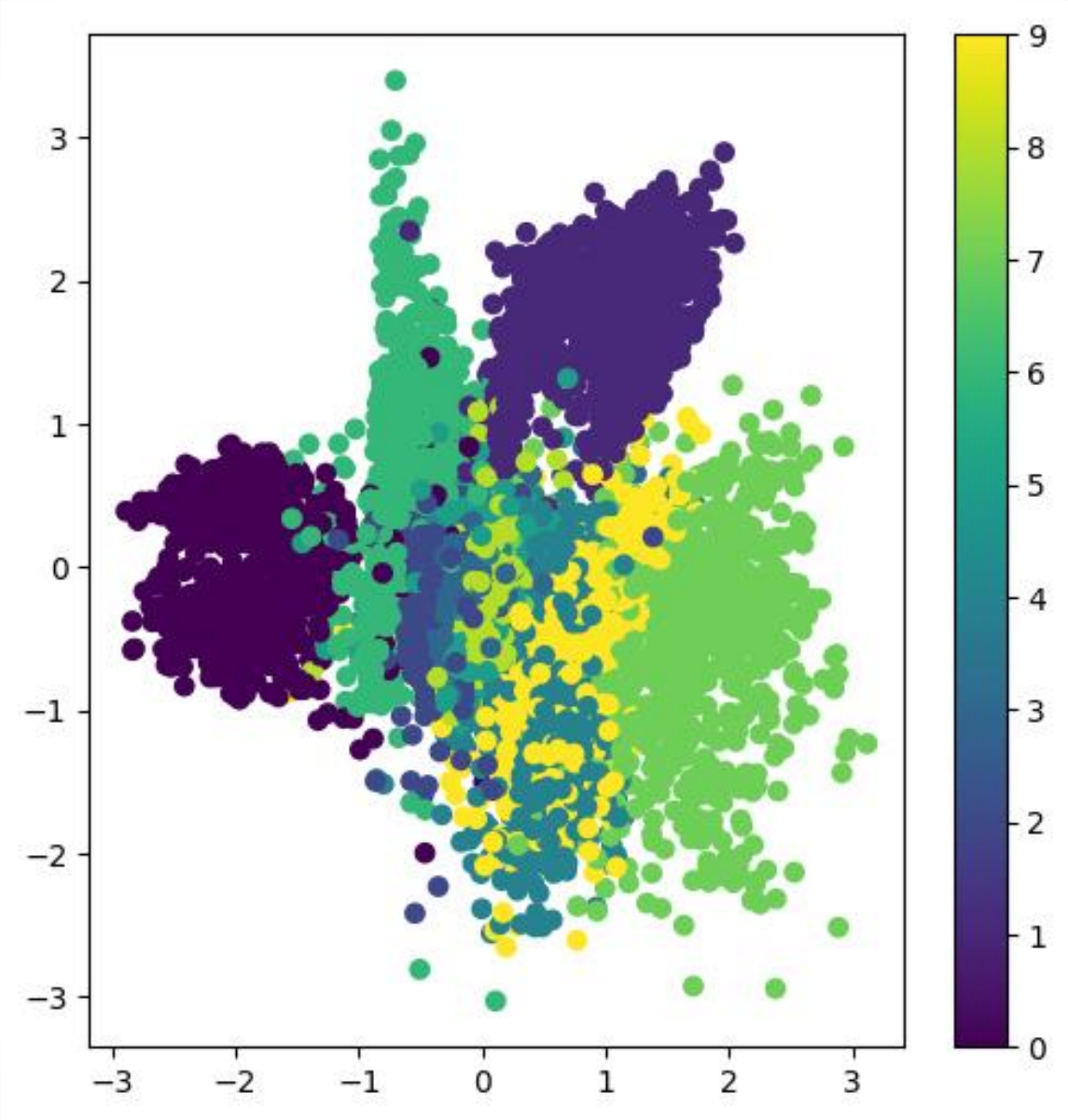
Auto-Encoder su MNIST

- SSIM 0,99
- PSNR 65,75



Auto-Encoder su CIFAR10

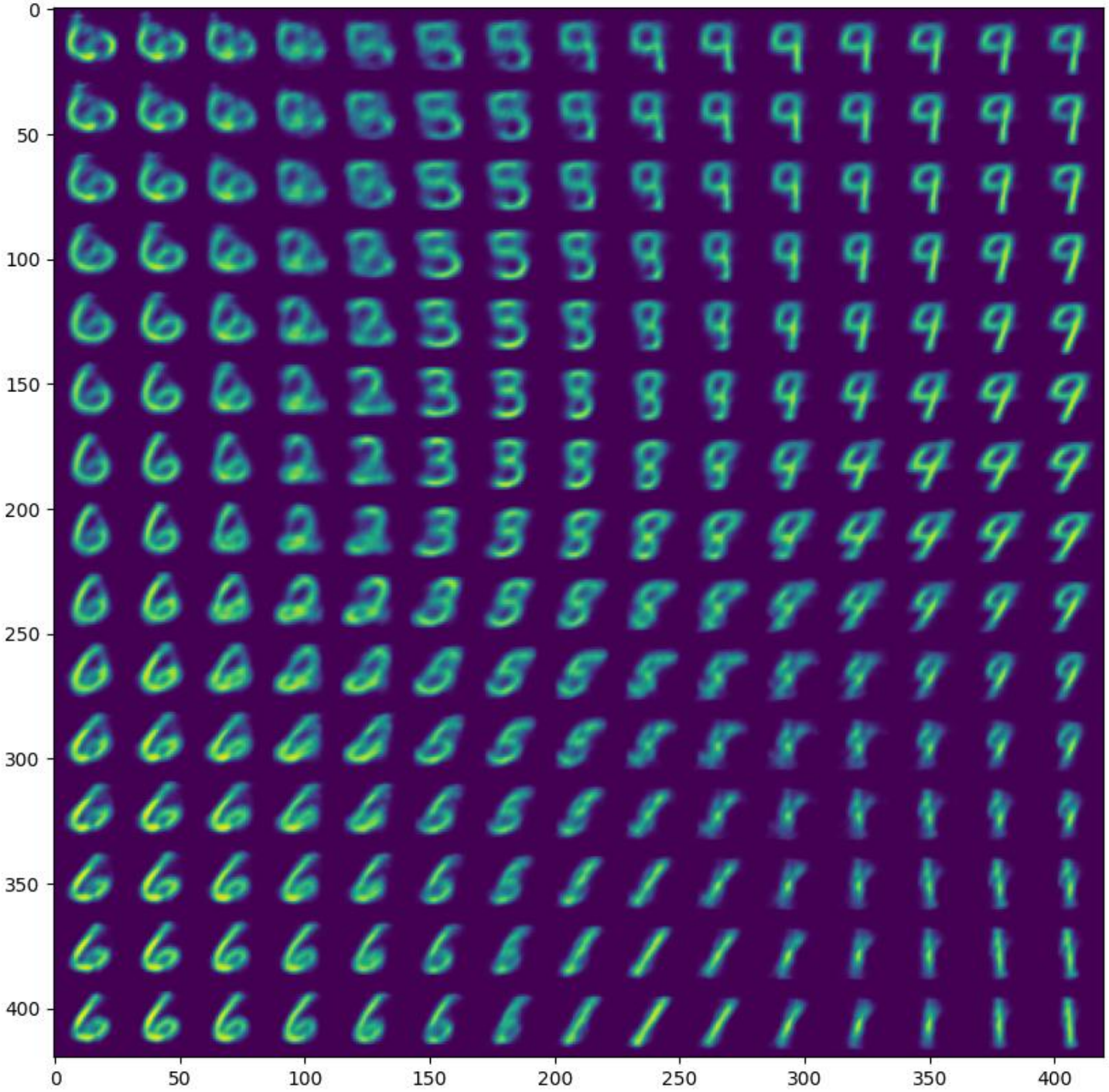
- SSIM
- PSNR



Variational Auto-Encoder su MNIST

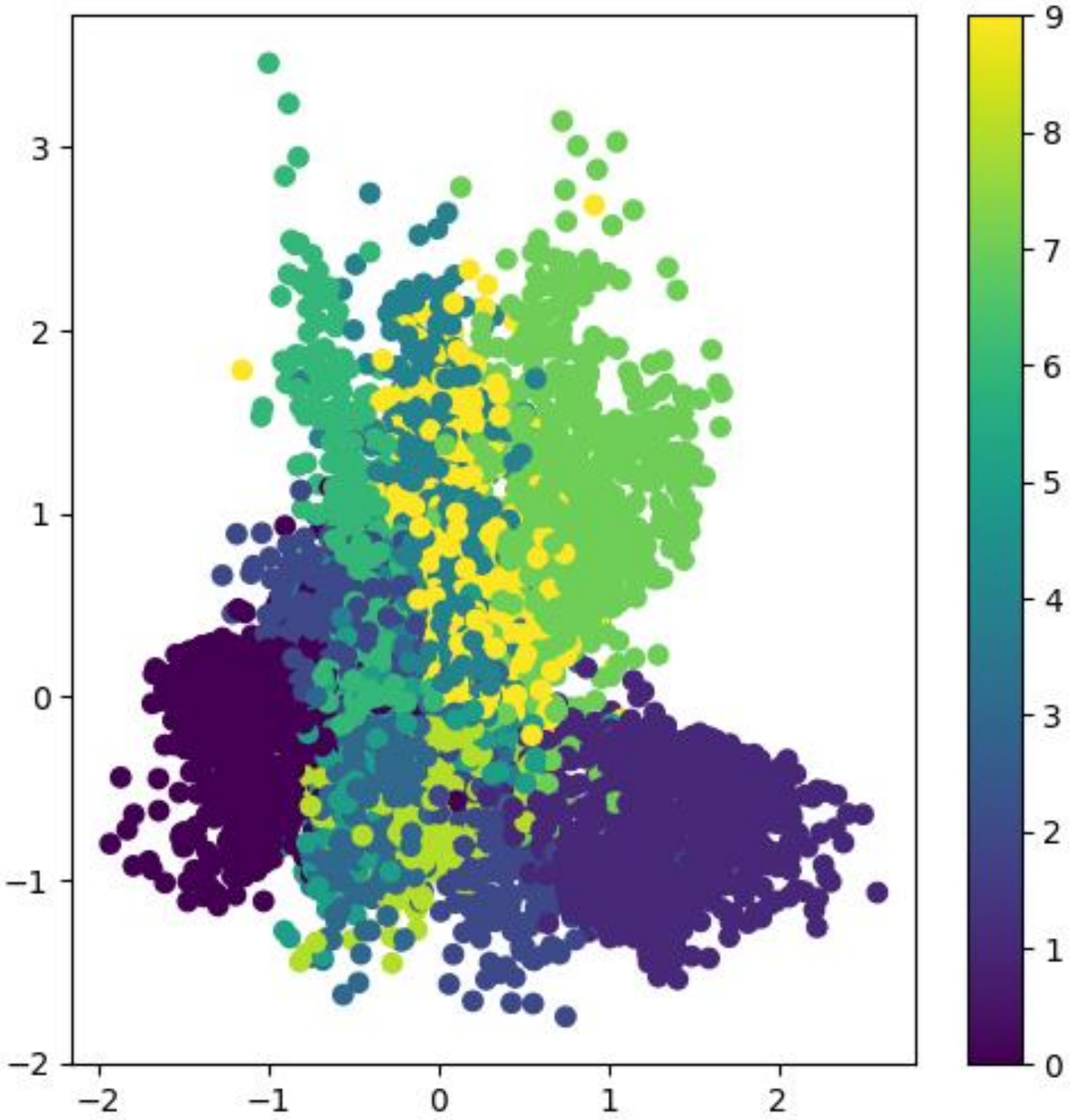
Caso di modello addestrato solo per l’encoding

- PSNR 62,57
- SSIM 0,99



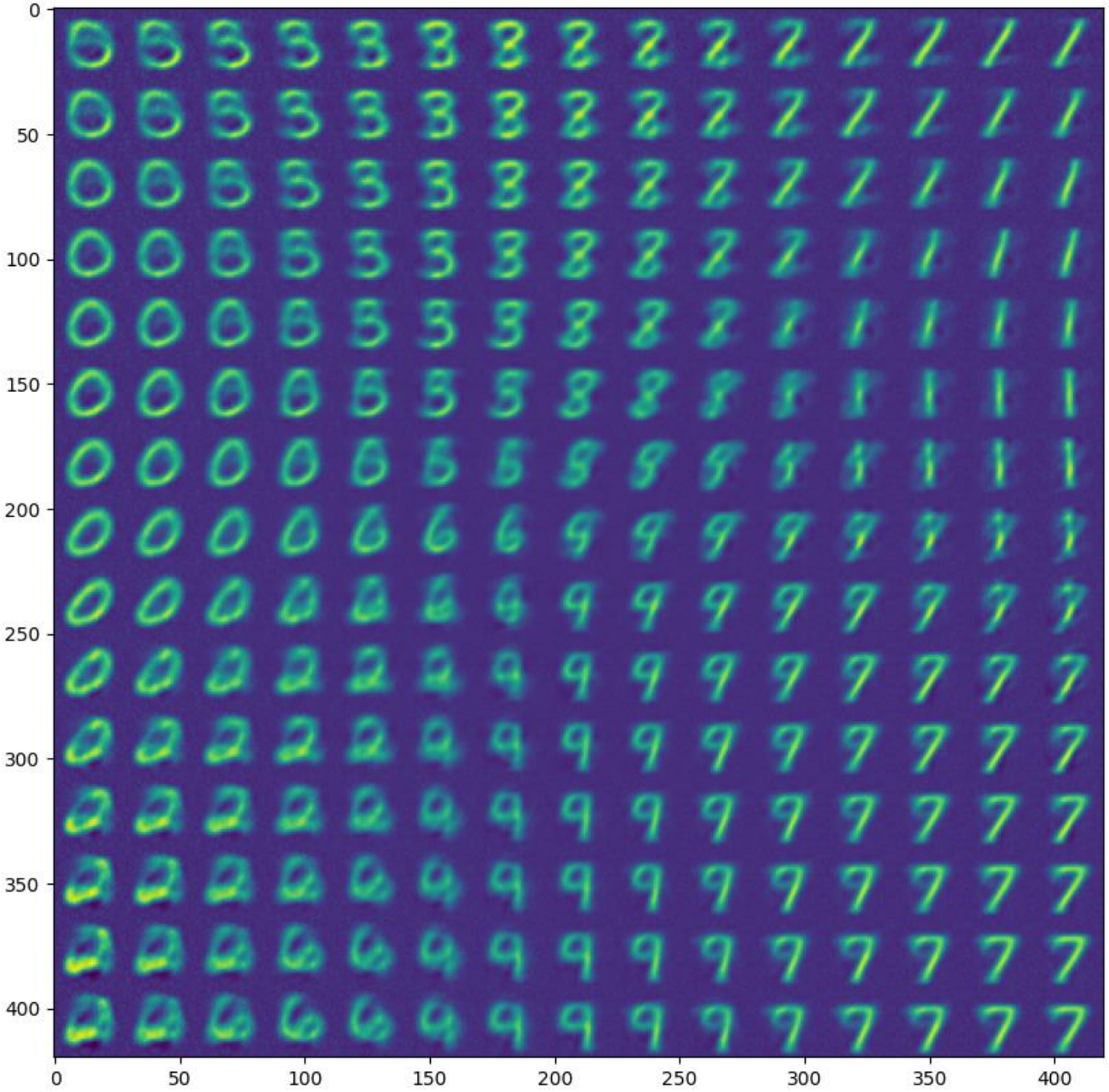
Variational Auto-Encoder su MNIST

- Caso di modello addestrato solo per l’encoding
- Punti decodificati presi da griglia uniforme



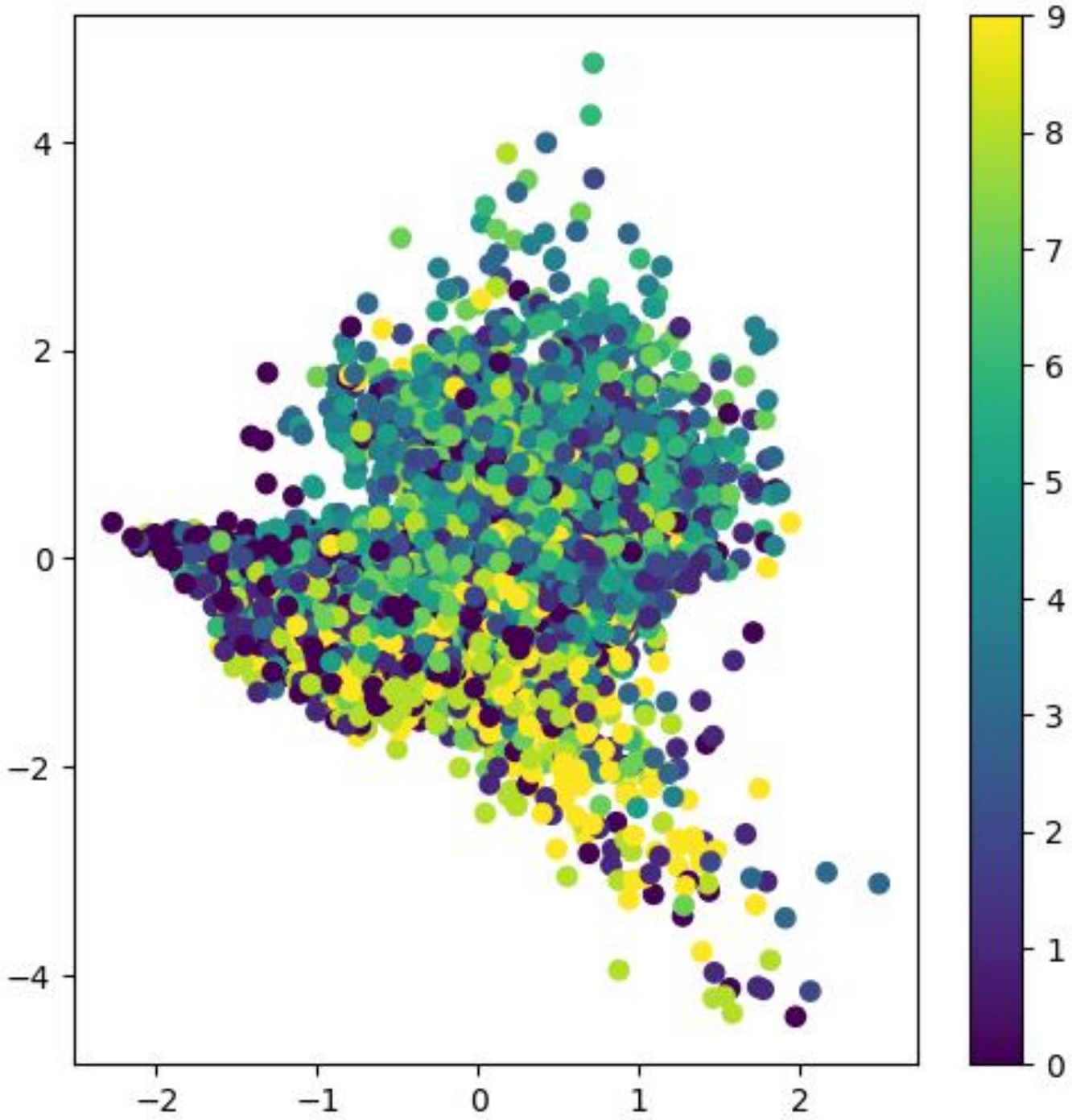
Variational Auto-Encoder su MNIST

- Immagini codificate nel latent space
- Addestrato per denoising
- SSIM 0,99
- PSNR 60,15



Variational Auto-Encoder su MNIST

Punti decodificati presi da griglia uniforme



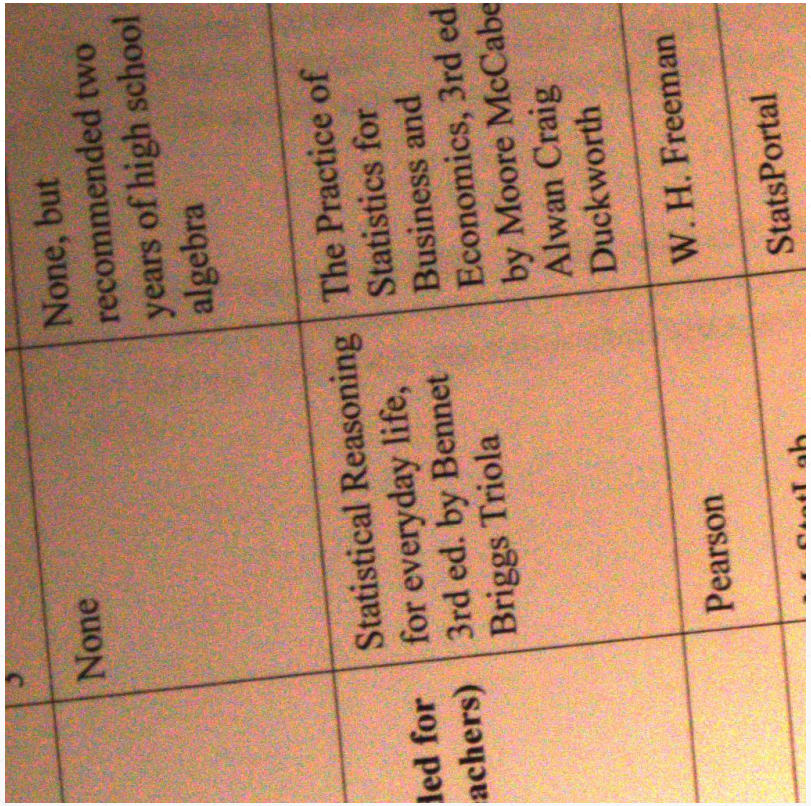
Variational Auto-Encoder su CIFAR10

Capacità di discriminazione del modello insufficiente

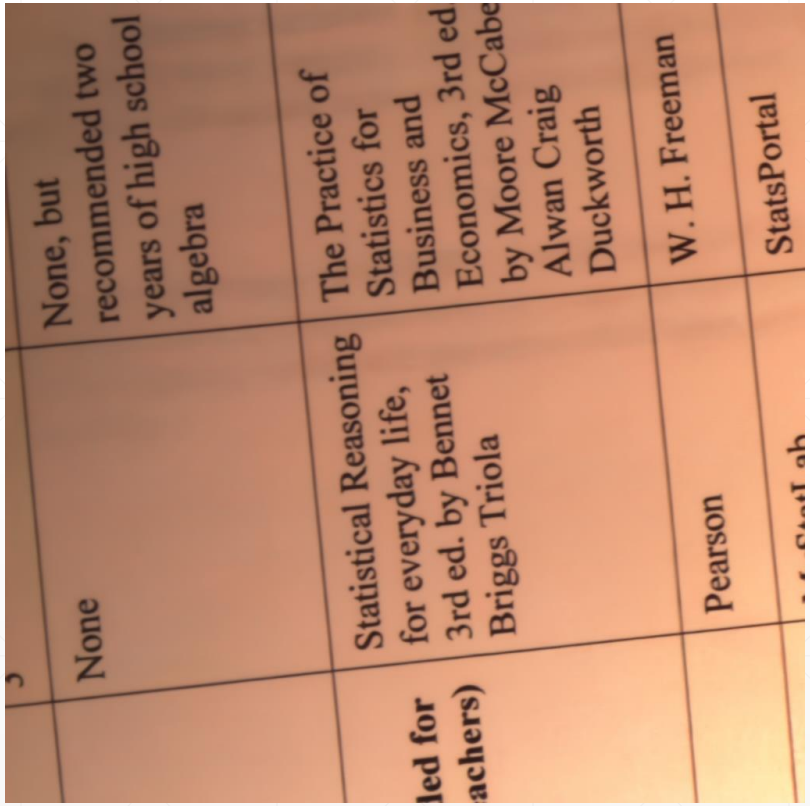
Restormer su Renoir

- SSIM 0,83
- PSNR 32,74

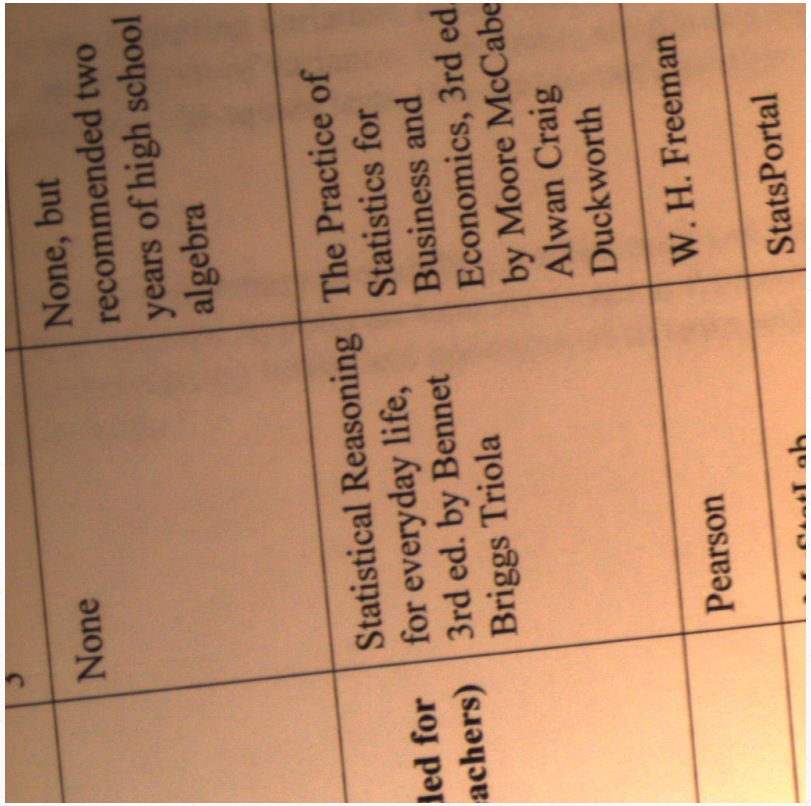
Restormer su Renoir



None	Statistical Reasoning for everyday life, 3rd ed. by Bennet Briggs Triola	The Practice of Statistics for Business and Economics, 3rd ed by Moore McCabe Alwan Craig Duckworth
None, but recommended two years of high school algebra	Pearson	W. H. Freeman
		StatsPortal



None	Statistical Reasoning for everyday life, 3rd ed. by Bennet Briggs Triola	The Practice of Statistics for Business and Economics, 3rd ed by Moore McCabe Alwan Craig Duckworth
None, but recommended two years of high school algebra	Pearson	W. H. Freeman
		StatsPortal



None	Statistical Reasoning for everyday life, 3rd ed. by Bennet Briggs Triola	The Practice of Statistics for Business and Economics, 3rd ed by Moore McCabe Alwan Craig Duckworth
None, but recommended two years of high school algebra	Pearson	W. H. Freeman
		StatsPortal

Metodi tradizionali

- Filtro Gaussiano
  - PSNR 11,99 SSIM 0,21
- Filtro mediano
  - PSNR 11,76 SSIM 0,20
- Filtro bilaterale
  - PSNR 11,79 SSIM 0,21

Grazie per l’attenzione