

Rete Convolutionale per Image Deblurring

Gruppo

June 4, 2025

Abstract

L'obiettivo del progetto è l'applicazione di una U-Net Convolutionale al fine di migliorare la qualità dell'immagine in input rimuovendo il Blur causato dal moto del soggetto acquisito (*Motion Blur*) o causato dalla messa a fuoco dell'obiettivo (*Focus Blur*)

1 Introduction

[1]

2 Theory and Traditional Approach

Una immagine con Blur è modellata matematicamente come convoluzione tra ground-truth image latente e blur kernel, dove si quest'ultimo essere *shift-invariant*. In questo caso, l'estrazione dell'immagine sharp è un problema di *Image Deconvolution*, la quale è suddivisa in *Non-blind-deconvolution* e *Blind-deconvolution*.

Formulazione Matematica:

$$\mathbf{b} = \mathbf{i} * \mathbf{k} + \mathbf{n}$$

Dove:

\mathbf{b} : Immagine con blur

\mathbf{i} : Immagine *ground-truth* latente

\mathbf{k} : Blur Kernel

\mathbf{n} : Rumore presente nell'immagine per contare imperfezioni causate dall'acquisizione (quantizzazione, saturazione del colore, risposta non lineare della camera, ...) (Esempio: rumore gaussiano)

Non-Blind Deconvolution In questa metodologia tradizionale, il blur kernel è noto a priori (Esempio: Point Spread Function Gaussiana per Blur senza direzione, Linea con direzione e lunghezza per Blur con direzione).

Uno dei primi metodi utilizzati in questa categoria, implementato come comparazione, è la *Wiener Deconvolution*, il cui obiettivo è la ricerca di un filtro \mathbf{g} tale che, tramite convoluzione con l'immagine blurred \mathbf{b} . Espresso nel dominio di Fourier:

$$\hat{\mathbf{I}} = \mathbf{G}\mathbf{B} \quad (1)$$

$$\mathbf{G} = \frac{|\mathbf{K}|^2}{|\mathbf{K}|^2 + \frac{1}{\text{SNR}}} \frac{1}{\mathbf{K}} \quad (2)$$

Dove:

\mathbf{G} e \mathbf{K} : trasformate di Fourier di \mathbf{g} e \mathbf{k}

SNR: Signal to noise ratio (infinitamente alto se rumore assente)

Un'implementazione di tale metodo di Deblurring si basa su un metodo di ottimizzazione convessa chiamato *Alternating Direction Method of Multipliers* (ADMM)¹

Blind Deconvolution In questa metodologia, il blur kernel è ignoto², dunque parte dell'algoritmo è la *PSF estimation*, modellata come stima di una stima di densità di probabilità

3 Architettura del modello

L'architettura utilizzata corrisponde in buona parte a quella illustrata in [1]. La rete si basa su di una struttura a U (encoder-decoder convolutionali), con estrazione delle feature effettuata a risoluzioni multiple (downsampled) e con l'aggiunta dei cosiddetti *Multi-Scale Modules* (MSM), che hanno l'obiettivo di implementare meccanismi di attenzione di diversa forma (quadrata e rettangolare).

In figura 1 è riportata l'architettura completa. La prima caratteristica interessante è che l'immagine in input viene elaborata non solo alla risoluzione primaria (256x256) generata dal dataloader, ma viene ulteriormente sottocampionata (rispettivamente alla metà e a un quarto della risoluzione) e reinserita nella rete come input dei layer successivi. L'obiettivo di questa configurazione multi-input multi-output è di analizzare l'immagine degradata secondo diversi livelli di dettaglio, e quindi di individuare pattern e feature più variegati e di diversa intensità.

L'encoder e il decoder hanno una struttura pressoché speculare, con tre skip connection che collegano i due rami, corrispondenti alle tre differenti risoluzioni a cui viene elaborata l'immagine. Come avviene di consueto nelle reti convolutive, al ridursi della dimensione del tensore in larghezza e altezza, cresce il nu-

¹https://stanford.edu/class/ee367/reading/lecture6_notes.pdf

²I metodi con neural network rientrano in questa categoria

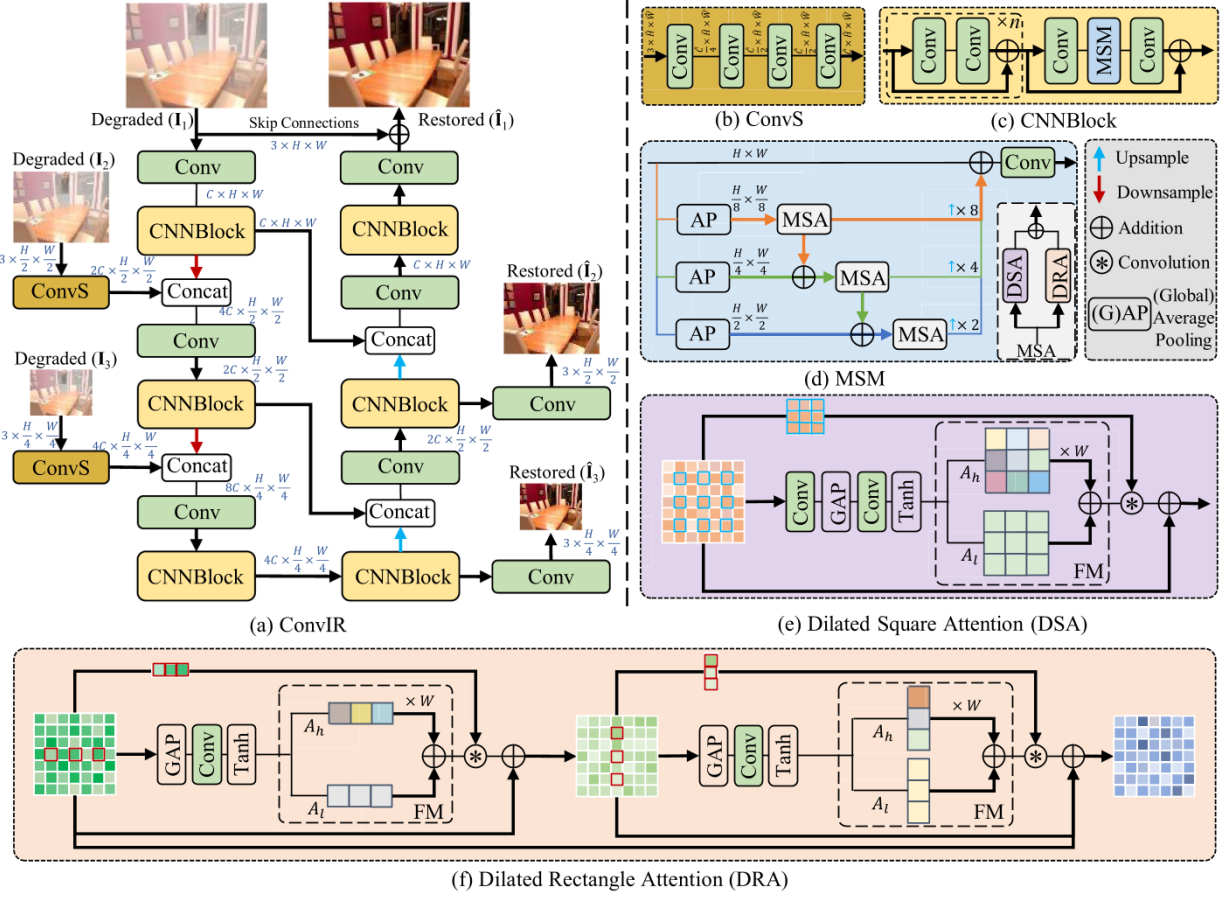


Figure 1: Schema generale dell'architettura completa (ConvIR)

mero di canali. La feature extraction passa infatti attraverso i seguenti moduli:

- layer convolutivo semplice (blocco *Conv* in verde in figura);
- ConvS*, blocco utilizzato solo per le versioni sottocampionate dell'immagine in input: consiste in una sequenza di quattro layer convolutivi che mantengono costanti le dimensioni di larghezza e altezza;
- CNNBlock*, blocco costituito da una serie di layer convolutivi raggruppati in $n+1$ blocchi residuali; nell'ultimo di questi blocchi viene inserito anche il *MSM*;
- MSM* (*Multi-Scale Module*): fonde l'elaborazione di tre moduli *MSA* (*Multi-Scale Attention*), che operano appunto su tre scale dimensionali gradualmente decrescenti. Ogni *MSA* combina di fatto l'output di un *DSA* e un *DRA*;
- DSA* (*Dilated Square Attention*): produce prima una attention map concentrandosi sulle aree quadrate del tensore in input, attraverso layer convolutivi e di pooling, e in seguito la elabora attraverso un filtro passa alto con parametri allenabili che sottrae unicamente la componente continua e tende ad esaltare quelle a più alta frequenza, tipicamente responsabili del blur;
- DRA* (*Dilated Rectangle Attention*): modulo analogo al precedente ma focalizzato su pattern

di forma rettangolare, combina attention map in senso verticale e orizzontale.

L'architettura originale proposta in [1] è stata utilizzata dagli autori in tre diversi formati: *S* (*Small*), *B* (*Base*) e *L* (*Large*), ognuna caratterizzata da un numero variabile di blocchi residuali all'interno di ciascun modulo *CNNBlock*. Le configurazioni prevedevano $n = 3$, $n = 7$ e $n = 15$ rispettivamente per le varianti *S*, *B* e *L*. La nostra scelta è ricaduta in una variante intermedia tra la *S* e la *B*, con $n = 6$ blocchi residuali. Abbiamo inoltre modificato il *CNNBlock* rimuovendo completamente il modulo *MSM* al suo interno. In questo modo, il training è risultato molto più rapido ed efficiente con solo una minima perdita di qualità del risultato finale.

La loss function utilizzata corrisponde alla somma pesata di un contributo calcolato nel dominio spaziale (\mathcal{L}_1) e uno in frequenza (\mathcal{L}_{freq}), in modo da considerare adeguatamente i diversi apporti dovuti alla presenza del blur:

$$\mathcal{L}_1 = \mathcal{L}_{spatial} = \sum_{i=1}^3 \frac{1}{P_i} \left\| \hat{\mathbf{I}}_i - \mathbf{Y}_i \right\|_1,$$

$$\mathcal{L}_{freq} = \sum_{i=1}^3 \frac{1}{S_i} \left\| [\mathcal{R}(\hat{\mathbf{I}}_i), \mathcal{I}(\hat{\mathbf{I}}_i)] - [\mathcal{R}(\mathbf{Y}_i), \mathcal{I}(\mathbf{Y}_i)] \right\|_1,$$

dove i indicizza gli output multipli a diverse risoluzioni; $\hat{\mathbf{I}}$ e \mathbf{Y} rappresentano rispettivamente

l'immagine elaborata dalla rete e il ground truth; P e S indicano il numero totale di elementi dei tensori presi in considerazione, in modo da avere delle metriche normalizzate; gli operatori $\mathcal{R}()$ e $\mathcal{I}()$ estraggono rispettivamente la parte reale e immaginaria della FFT operata sull'immagine.

La funzione di costo complessiva è così calcolata:

$$\mathcal{L}_{tot} = \mathcal{L}_{spatial} + \lambda \mathcal{L}_{freq},$$

dove λ è un iperparametro impostato di default a 0.01.

4 Observations

5 Results

Glossary

blind-deconvolution Metodo di estrazione di ground-truth image in cui il blur kernel è ignoto. 1

non-blind-deconvolution Metodo di estrazione di ground-truth image in cui il blur kernel è noto. 1

shift-invariant Un kernel è shift-invariant se e solo se la sua trasformata di Fourier non cambia con la traslazione del contenuto dell'immagine. 1

References

- [1] Yuning Cui et al. "Revitalizing Convolutional Network for Image Restoration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.12 (2024), pp. 9423–9438. DOI: 10.1109/TPAMI.2024.3419007.