

1. Scelte di progetto

Ho scelto di usare Matlab e il suo pacchetto image processing toolbox. La scelta deriva dalla flessibilità garantita dall'applicazione e dalle sue grandi capacità di calcolo. Per una visualizzazione migliore dei risultati delle esecuzioni è stato usato il live editor che permette di vedere l'output di un'istruzione sotto l'istruzione stessa.

2. Istruzioni per esecuzione

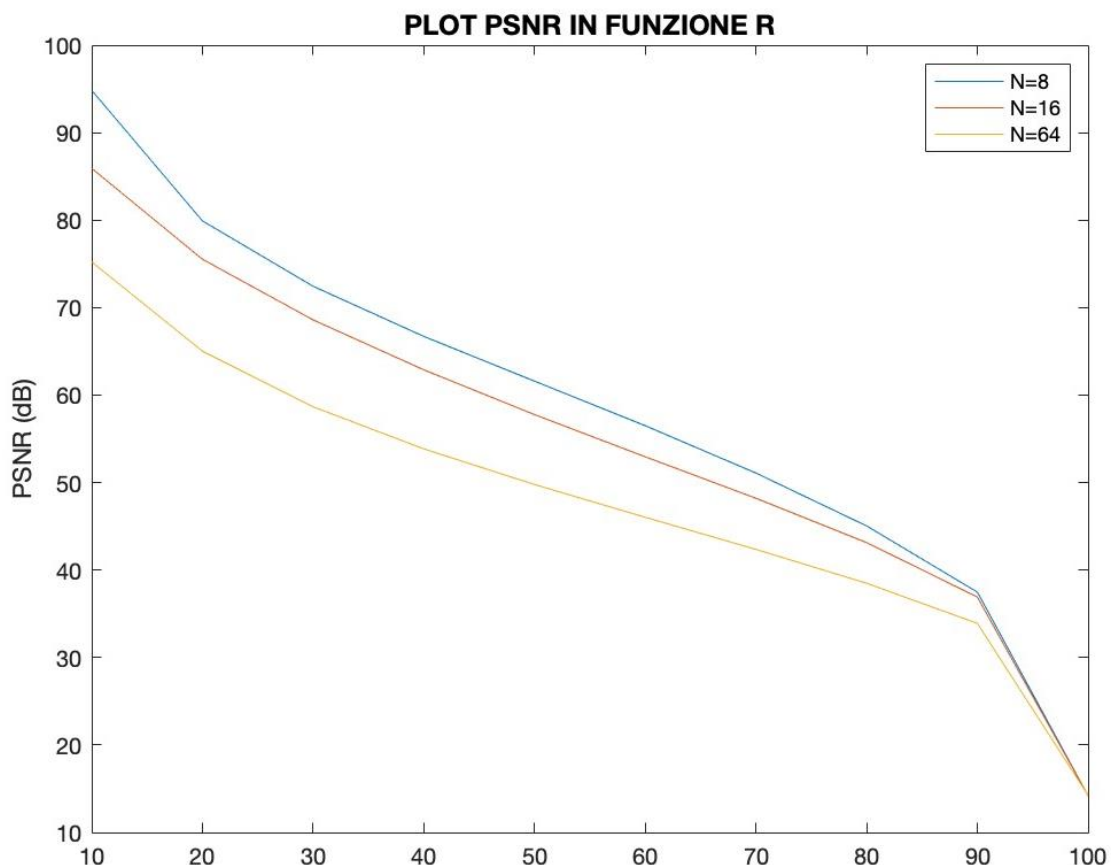
L'esecuzione del codice è stata effettuata sul sistema operativo MacOS Monterey con MATLAB_R2023a. Per eseguire il file Matlab serve posizionarlo nella stessa cartella dell'immagine che si vuole analizzare. Inoltre, bisogna selezionare tale cartella come cartella di lavoro. A questo punto si sceglie l'immagine da analizzare modificando la stringa passata come parametro a `imread()`. Il programma è stato testato solo con immagini in formato bmp ma dovrebbe funzionare anche con immagini in formato jpeg e png. In caso di errori nella lettura dell'immagine viene lanciato un errore e il programma termina. Il codice è commentato passo passo nel file `MatLab(.mlx)`.

3. Esempi codifica e commenti

Il programma è stato eseguito con due immagini di prova (`coffee.bmp`, `colors.bmp`), entrambe presenti nella cartella. I risultati delle esecuzioni sono i grafici presentati qui di seguito.

Guardando i grafici qui presentati possiamo vedere e confermare che il PSNR decresce all'aumentare di N (a parità del fattore $R\%$); infatti, con $N=8$ abbiamo il valore massimo di PSNR, mentre $N=64$ abbiamo il valore minimo di PSNR, questo deriva dal fatto che se abbiamo blocchi più grandi abbiamo una distorsione maggiore. Inoltre, possiamo vedere come il PSNR decresca mentre R aumenta dato uno specifico valore di N , questo perché il valore di soglia sarà più alto e quindi si tradurrà in numero di valori azzerati maggiore.

-Colors.bmp:



-Coffee.bpm:

