SCELTE DI PROGETTO

Per questo homework è stato scelto come linguaggio Python. La sua proprietà di linguaggio ad alto livello permette di affidarsi ad una grande varietà di strutture dati e moduli già definiti nella scrittura di un codice efficace, che possa rappresentare i passaggi teorici dell'algoritmo di codifica e allo stesso tempo comprimere i calcoli.

È stata quindi usata una varietà di librerie:

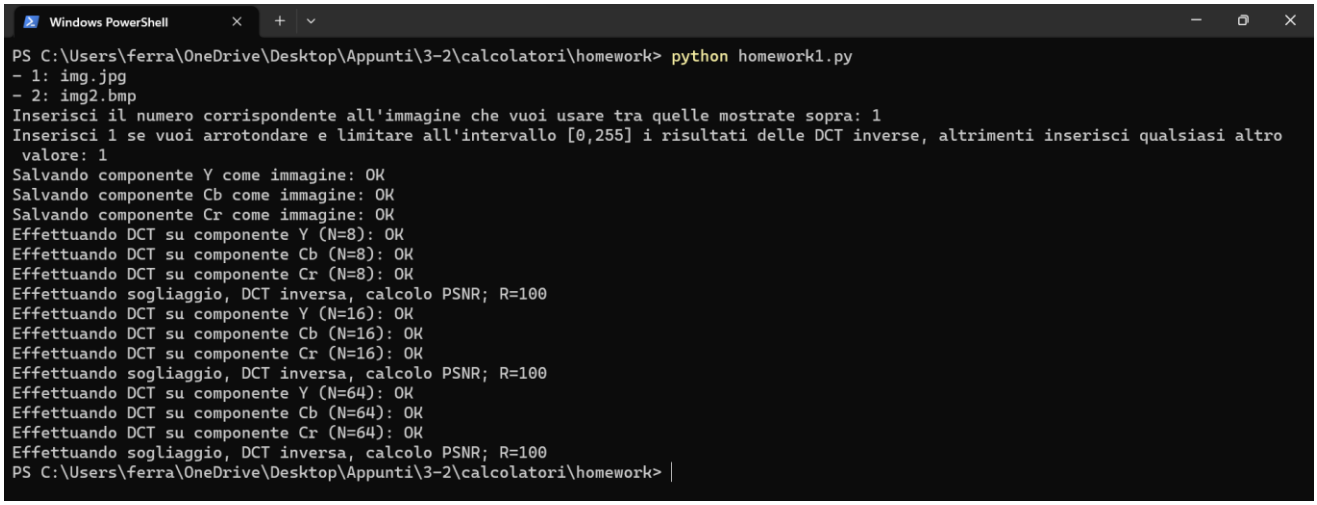
- *PIL (Python Imaging Library)*: usata per aprire e manipolare le immagini ed estrarne le componenti YCbCr
- *NumPy*: usata per lavorare in maniera compatta ed efficiente sulle varie matrici delle componenti dell'immagine, in tutte le fasi dell'algoritmo
- *SciPy*: importata per usare DCT e DCT inversa bidimensionali
- *Matplotlib*: usata per calcolare e visualizzare i grafici del PSNR, tramite la sua interfaccia *pyplot*
- *os:* usata per ottenere la lista di immagini presenti nella cartella corrente

È stato scelto inoltre di fornire un semplice script Python al posto di un Jupyter notebook per rendere più snello il codice e la sua esecuzione, comunicando all'utente i messaggi essenziali sul proseguimento del codice.

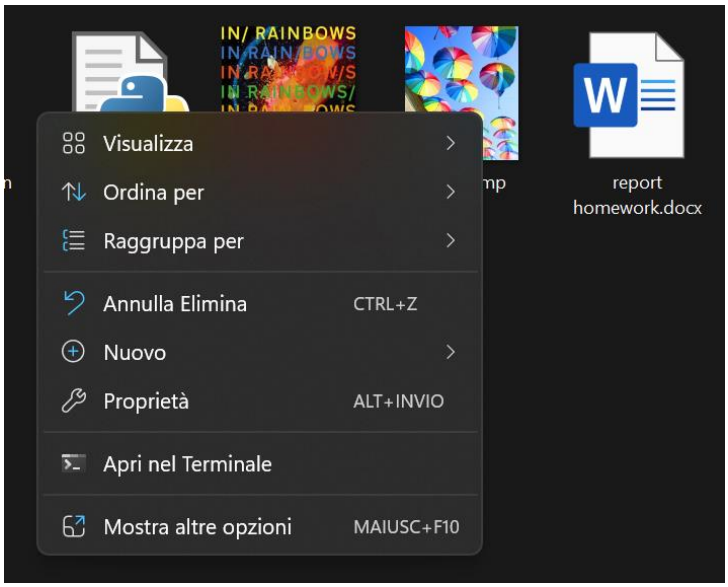
ISTRUZIONI DI ESECUZIONE

Il primo passaggio consiste nell'installazione dei moduli sopra citati. Il modo più semplice per installarli è aprire il prompt dei comandi di Windows (Win+R -> digitare "cmd" nella finestra appena aperta -> Invio) e usare pip: *pip install numpy*, *pip install pillow* (fork di PIL), *pip install scipy*, *pip install matplotlib*. Si consiglia di installarli nell'ordine indicato, dato che Scipy richiede la presenza di NumPy e Matplotlib richiede sia NumPy che Pillow (PIL). Il modulo *os* è preinstallato con Python, quindi non serve installarlo manualmente.

Per eseguire il codice il metodo consigliato è usare il prompt dei comandi di Windows (si può anche eseguire dall'IDLE di Python, ma il carriage return usato in alcuni print del codice non funzionano e tutte le righe "Effettuando sogliaggio…" vengono stampate attaccate). Una volta aperto, navigare fino alla cartella in cui è stato estratto il contenuto del file zip ed eseguire il comando *python homework1.py* (su Windows) / *python3 homework1.py* (su Linux).



Su Windows 11, è possibile anche aprire la cartella con Esplora Risorse, fare clic destro sullo spazio vuoto (non su un file) e scegliere "Apri nel terminale".

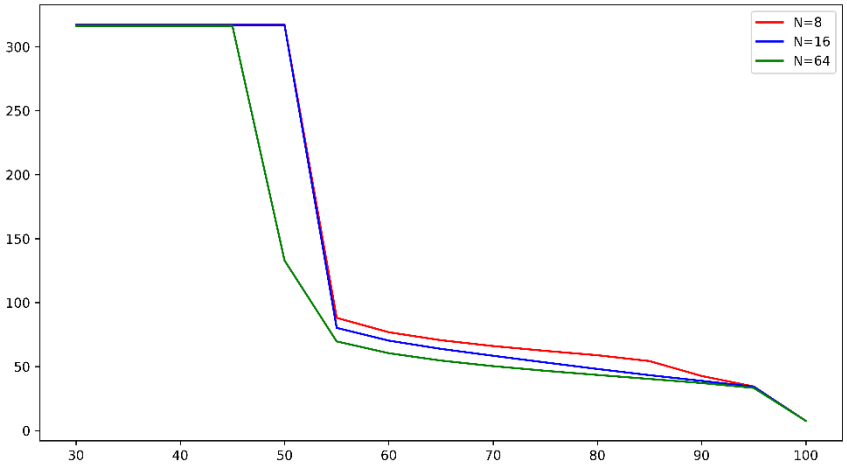


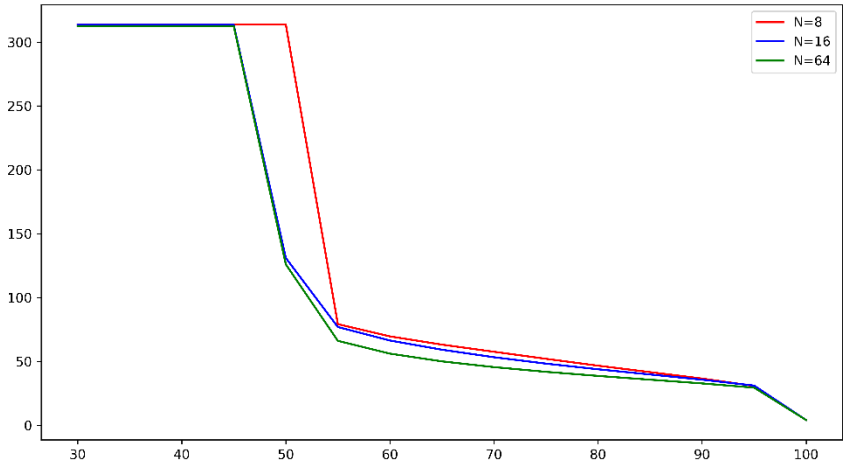
Alla fine del codice, viene aperta una finestra con il grafico calcolato da Matplotlib; una volta chiusa, l'esecuzione del codice termina. Nella cartella del file vengono salvate le componenti Y, Cb e Cr dell'immagine originale visualizzate in scala di grigi e il grafico del PSNR in funzione di R e N.

ESEMPI DI CODIFICA

Il programma è stato testato con due immagini diverse; la prima ha una risoluzione 1000x1000 ed è salvata in formato JPG, la seconda ha una risoluzione 400x472 ed è salvata in formato BMP:

Ecco le curve generate:



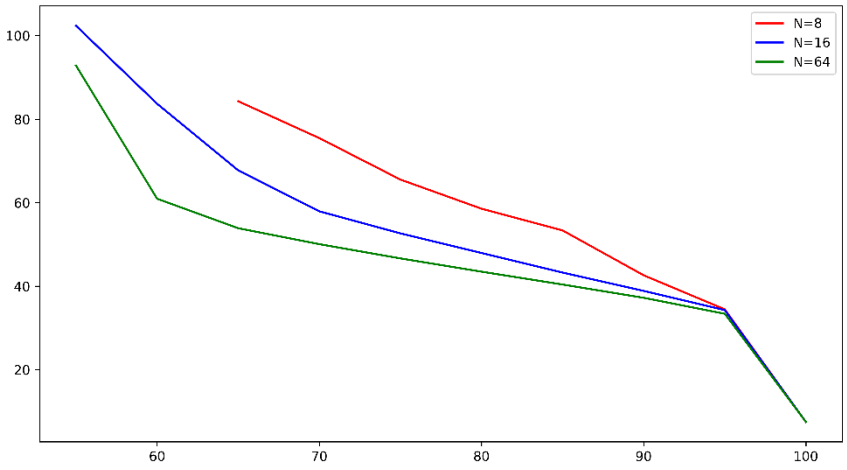


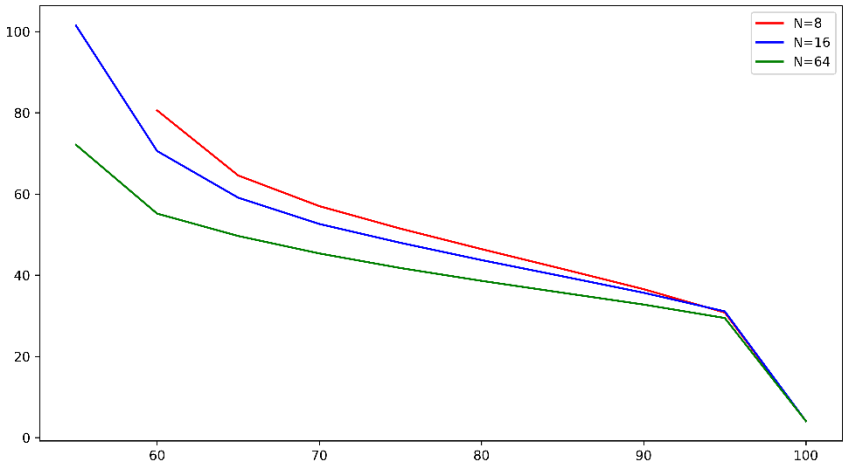
In entrambe si manifesta un comportamente simile. Per valori bassi di R, ovvero azzerando pochi coefficienti, il PSNR assume valore alto. Ciò segnale una distorsione quasi nulla, risultato ottenuto tramite una compressione molto debole.

Per valori di R maggiori di 40, il PSNR subisce una brusca riduzione; la distorsione comincia a diventare non trascurabile. Inoltre, questo succede per valori di R che diminuiscono all'aumentare di N. Questo indica che blocchi più grandi provocano una compressione e una distorsione maggiori.

Infine, dopo avere un tasso di decrescita costante per R e N, il PSNR subisce un altro brusco calo. Per valori di R maggiori di 95, quindi solo per valori molto grandi, la distorsione comincia ad essere visibile sulle immagini compresse.

Come ulteriore osservazione, questi grafici sono stati ottenuti calcolando i valori del PSNR usando i dati grezzi restituiti dalla DCT inversa. Infatti, per un'eventuale ricostruzione dell'immagine originale, sarebbe necessario arrotondare i valori ritornati e limitarli all'intervallo [0,255], che corrisponde al range accettabile per i pixel di un'immagine espressa in YCbCr. Se eseguiamo queste operazioni, le curve cambiano:





I valori rappresentati partono da R compreso tra 50 e 60; questo perché per valori inferiori l'MSE pesato risulta nullo, il che causerebbe un PSNR dal valore infinito. Inoltre i valori risultano in generale superiori rispetto alle curve precedenti. Questo indica che eseguire arrotondamento e limitazione dei valori diminuisce la distorsione.