UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

A.A. 2023/2024
Laurea Magistrale
Informatica

Progetto 2

Metodi del Calcolo Scientifico – Giugno 2024

Realizzato da:

Biancini Mattia – 865966

Gargiulo Elio - 869184

Lo Scopo del Progetto

DCT2 Custom

Implementazione della **Discrete Cosine Transform 2** e comparazione con la versione offerta dalla libreria del linguaggio **Python.**

La comparazione è stata fatta a livello di tempi di **calcolo** e **correttezza**.

Compressione Immagini

Implementazione di un programma che permetta di poter comprimere delle immagini in formato **bitmap** e scala di grigio. Implementato in **Python.**

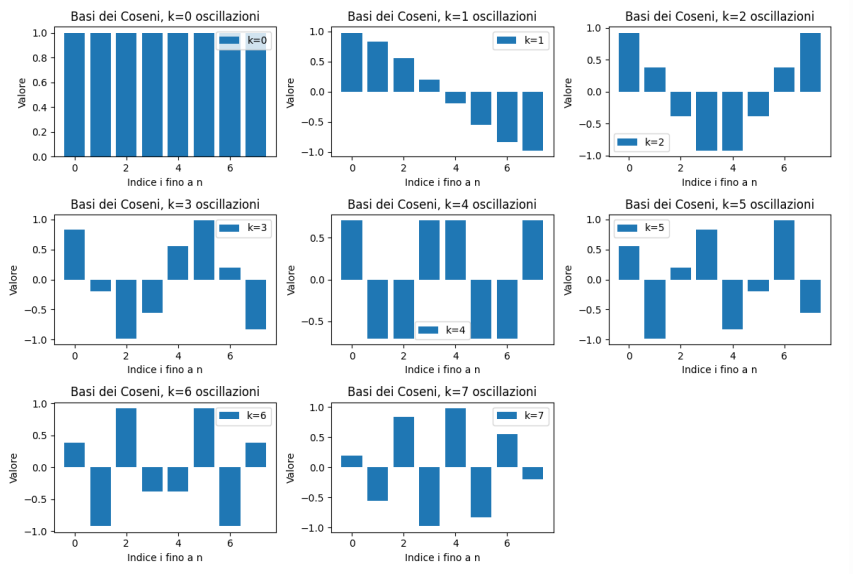
Il risultato verrà mostrato a schermo, affianco all'immagine originale.

Parte 1 - Implementazione della DCT2

L'approccio utilizzato è stato quello di implementare la DCT, per poi eseguirla su righe e colonne della matrice in input (DCT2).

Variazione al prodotto $W_k \times W_k$ per il calcolo dei coefficienti A_k:

$$A_k = \begin{cases} \sqrt{N} & se\ k = 0 \\ \sqrt{\frac{N}{2}} & se\ k > 0 \end{cases}$$



Plot della Base del Coseni, utilizzato per verificare la correttezza del calcolo dei W_k

Parte 1 - DCT2 della Libreria

La libreria scelta per la DCT2 è stata **SciPy** di Python.

Essa fornisce, oltre a molteplici funzioni per calcolo scientifico, diversi tipi di DCT2 tra cui quella mostrata nel corso.

I parametri utilizzati sono:

- Type 2
- Normalizzazione Ortogonale

Va notato che la versione della DCT2 di SciPy è la versione **Fast.**

Sarà quindi più efficiente di quella implementata a mano.

Parte 1 - Cosa ci si Aspetta?

DCT2 Custom

L'implementazione della nostra DCT2 sarà meno efficiente di quella della libreria, aggirandosi su tempi di calcolo proporzionali a N^3.

DCT2 Library

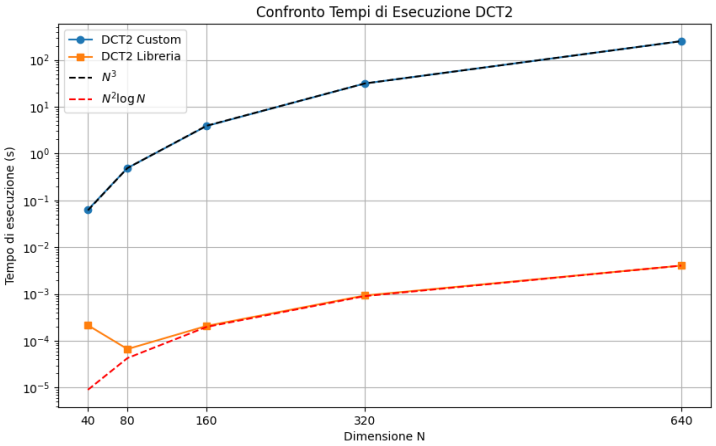
L'implementazione della DCT2 fornita da **SciPy** sarà sicuramente ottimizzata, utilizzando la Fast Fourier Transform. I tempi di calcolo si aggireranno intorno a $N^2 \log N$.

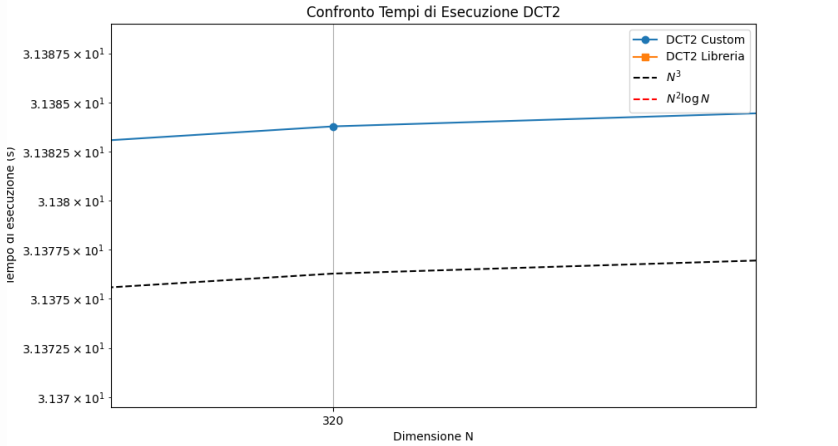
Valutazione

Al fine di calcolare i tempi, sono state generate delle matrici casuali a partire dalla dimensione **40x40** fino a **640x640**, raddoppiando di volta in volta.

I tempi verranno analizzati in un grafico a scala semilogaritmica.

Parte 1 - Risultati Ottenuti

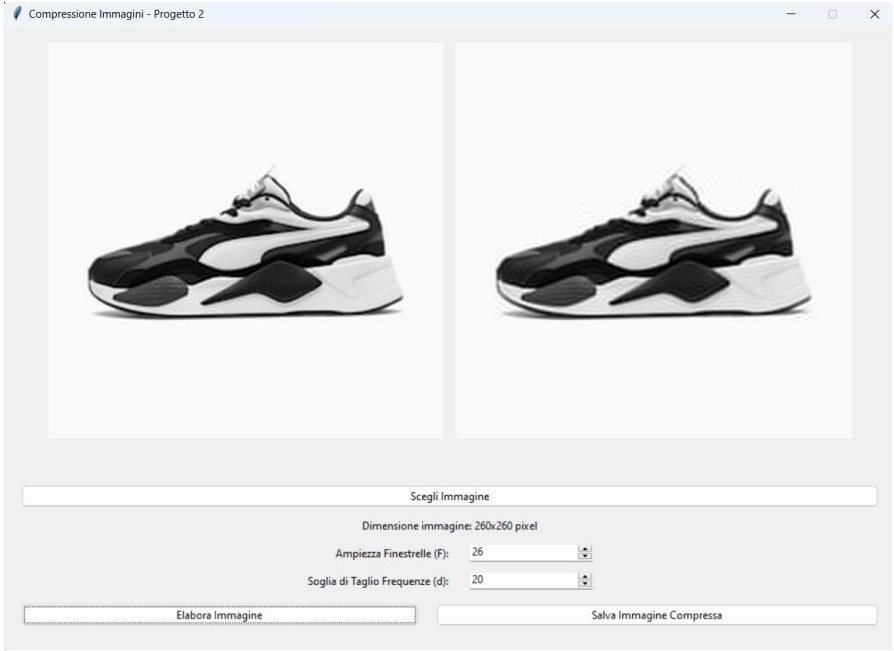




DCT2 Custom tende effettivamente a N^3, mentre SciPy utilizza la versione Fast.

Possiamo notare che comunque le due curve siano leggermente diverse se zoomate per la DCT2 Custom.

Parte 2 - Interfaccia Grafica



L'Interfaccia permette di:

- **Scegliere un'immagine in formato bitmap.**
- **Scegliere l'ampiezza delle finestrelle (blocchi) F.**
- **Scegliere la soglia di taglio delle frequenze d.**
- **Elaborare l'immagine una volta inserito i parametri corretti.**
- **Salvare, se si desidera, l'immagine elaborata.**

Parte 2 - Considerazioni e Casi

Sono stati analizzati diversi casi durante la fase di testing:

- **Casi Normali, con diverse qualità**
- **Fenomeno di Gibbs**
- **Scarti Notevoli**
- **Taglio delle Frequenze a 0**

In più è stato scelto, opzionalmente, di poter salvare l'immagine per un confronto più nel dettaglio.

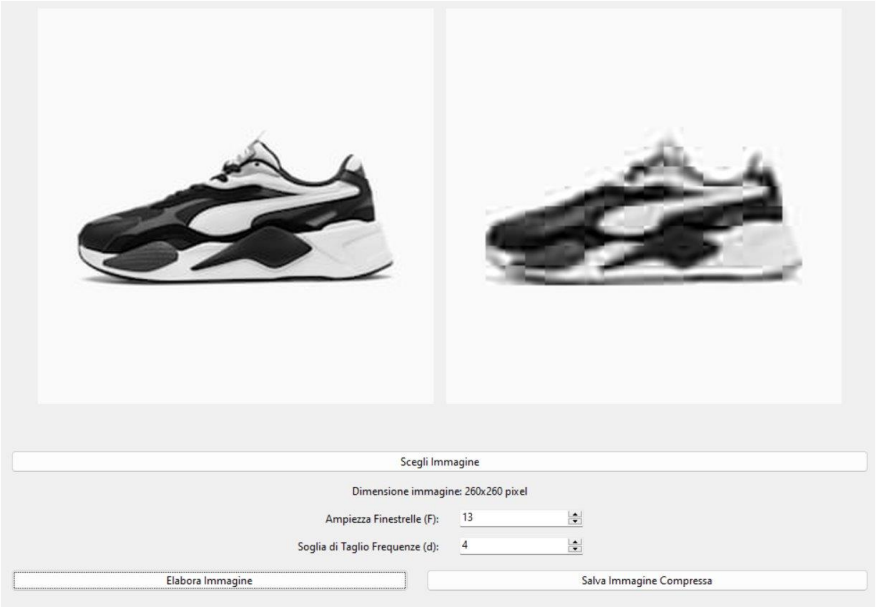
Le dimensioni non verranno impattate perché andrebbe salvata la compressione **pre-IDCT2**, se non per il salvataggio in 8 bit, scala di grigi di **bitmap.**

Parte 2 - Elaborazioni Generali

Le elaborazioni generali sono state concepite con lo scopo di verificare alterazioni nella qualità

Sono stati testati diversi parametri e vi è riportato il caso più visibile con una qualità molto bassa, **con un taglio delle frequenze aggressivo.**

Parametri: d = 4 e F = 13

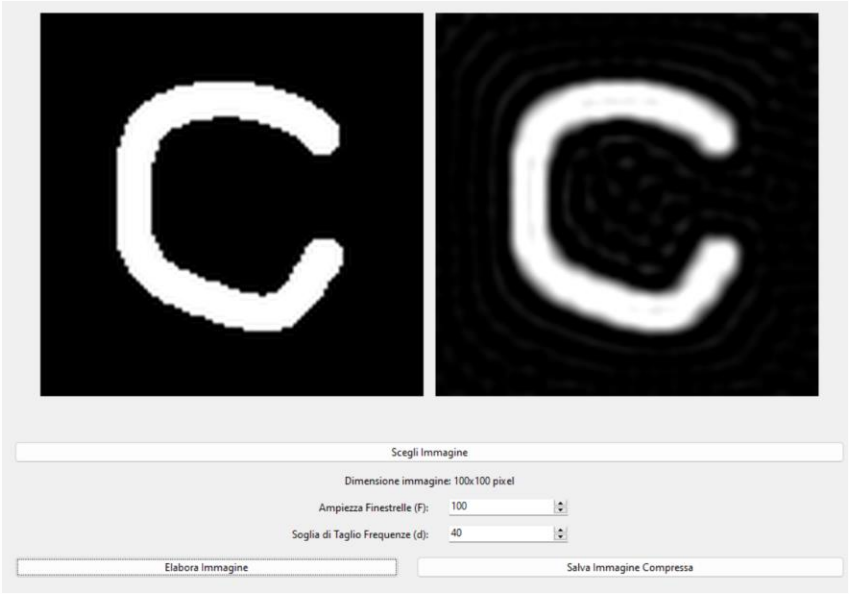


Parte 2 - Fenomeno di Gibbs

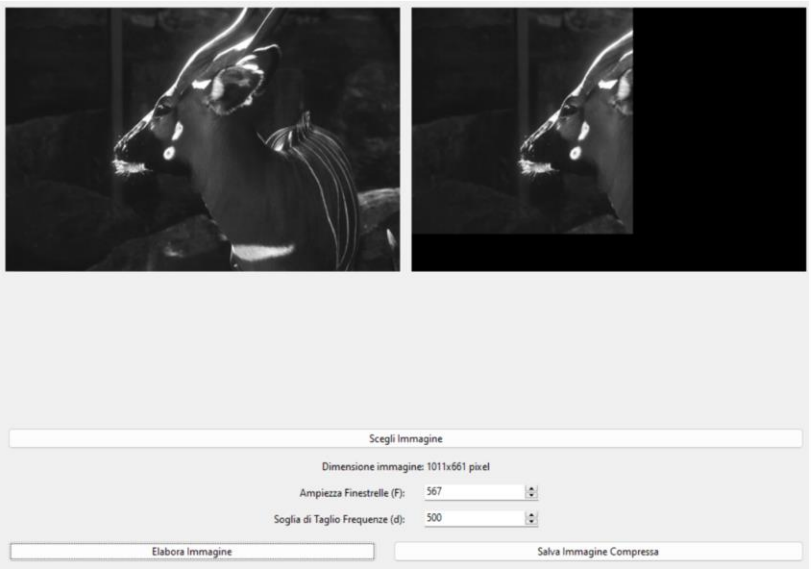
Replicato in modo simile a ciò è stato presentato nel corso, il Fenomeno di Gibbs si visualizza nelle immagini quando vi è una discontinuità netta (colori in questo caso).

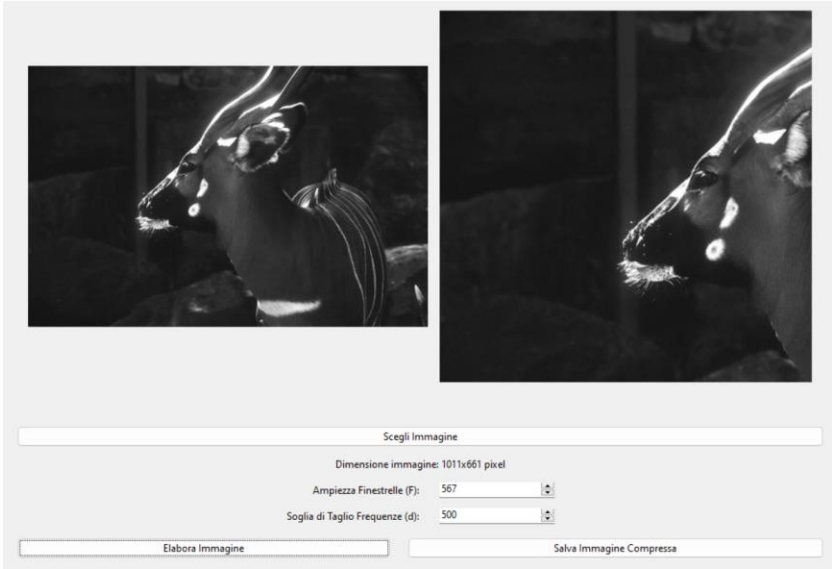
Per una resa evidente del fenomeno è stato scelto un F che fosse un solo blocco ed un taglio medio.

Parametri: d = 40 e F = 100



Parte 2 - Scarti Notevoli





Scegliendo i blocchi F non divisibili per la dimensione dell'immagine, l'immagine verrà tagliata. Questo può portare a dei grandi scarti.

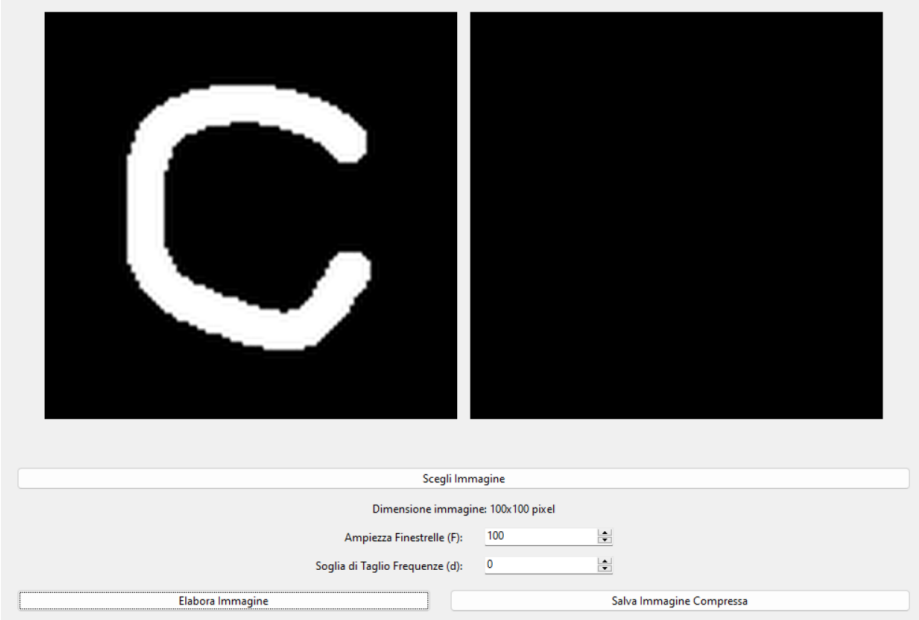
Parametri: d = 500 e F = 567

Parte 2 - Taglio delle Frequenze = 0

Tagliando tutte le frequenze, l'immagine ricostruita sarà logicamente vuota.

A livello di implementazione e analisi è stato scelto di mostrare un'immagine nera invece di un errore.

Parametri: d = 0 e F = 100



Parte 2 - Conclusioni Finali

Gli esperimenti effettuati, variando i parametri d ed F permettono di evidenziare vari effetti della compressione, introdotti a livello teorico nel corso.

Possiamo notare che con dimensioni dei blocchi molto grandi il fenomeno di Gibbs sia presente per distacchi di colori evidenti, confermando la necessità di blocchi di dimensione ridotta, oltre ad essere inefficiente.

La soluzione dell'algoritmo JPEG si rileva veramente efficace, rendendo poco percettibili i problemi derivati dagli scarti notevoli replicando i pixel ai bordi e attenuando il fenomeno di Gibbs con blocchi 8x8.