

UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Compressione di immagini tramite autoencoder, stato dell'arte e sviluppi futuri

Relatore

Prof. Cagnazzo Marco

Laureando

Stella Filippo

ANNO ACCADEMICO 2022-2023

Data di laurea GG/MM/AAAA

Ringraziamenti

Sommario

Questo elaborato esplora il campo emergente della compressione delle immagini utilizzando l'intelligenza artificiale. L'obiettivo principale è studiare e analizzare le tecniche esistenti per la compressione delle immagini, nello specifico quelle che fanno uso di autoencoder.

Nel primo capitolo, viene fornita una panoramica dello stato dell'arte della compressione delle immagini utilizzando le tecniche tradizionali. Viene fornita una rapida spiegazione del funzionamento di queste tecniche.

Il secondo capitolo si concentra sull'analisi delle tecniche per la compressione delle immagini tramite intelligenza artificiale. Viene fornita una rapida panoramica dell'attuale stato della ricerca. Successivamente viene fornita una spiegazione del principio di funzionamento degli autoencoder. Infine vengono presentate alcune ricerche che hanno portato un apporto significativo nella ricerca di queste tecniche.

Nel terzo capitolo vengono analizzate le prestazioni di alcuni metodi in termini di qualità oggettiva e soggettiva dell'immagine, compressione e tempo di compressione. Nel quarto e ultimo capitolo vengono esplorati i possibili sviluppi futuri nel campo della compressione delle immagini tramite autoencoder. Si discute di come le tecniche attuali potrebbero essere migliorate e si ipotizzano nuove direzioni di ricerca.

In conclusione, questa tesi fornisce una panoramica completa dello stato dell'arte della compressione delle immagini tramite autoencoder e offre spunti preziosi per future ricerche in questo campo.

Indice

Introduzione	1
1 Metodi tradizionali	5
1.1 JPEG	5
1.2 JPEG2000	5
1.3 BPG	6
1.4 VVC	6
2 Metodi con apprendimento automatico	9
2.1 Ballé 2018	9
2.2 Cheng 2020	9
2.3 Yang 2021	9
2.4 Wang 2022	9
3 Valutazione delle prestazioni	11
3.1 Metriche utilizzate	11
3.1.1 BPP	11
3.1.2 Tempo di codifica	11
3.1.3 PSNR	11
3.1.4 MSSIM	11
3.1.5 LPIPS	11
3.2 Presentazione dei risultati	11
4 Sviluppi futuri	13
4.1 Possibili ottimizzazioni	13
4.1.1 Spectral ADAM	13
4.2 Dispositivi mobili	13
4.2.1 Small CAE	13

5 Conclusioni **15**

Bibliografia **17**

Elenco delle figure

1	Diagramma di compressione Lossy	3
1.1	Confronto immagini PNG con JPEG bpp medio = 0.174	7
1.2	Confronto immagini PNG con JPEG2000 bpp medio = 0.171	7
1.3	Confronto immagini BPG con JPEG bpp medio = 0.179	8
1.4	Confronto immagini VVC con JPEG bpp medio = 0.164	8

Introduzione

Nell'era moderna la crescente quantità di dati di cui usufruiamo ogni giorno sta ricevendo, dagli esperti del settore, molte attenzioni. In quanto il throughput e la quantità di memoria di cui disponiamo sui nostri dispositivi sono, seppur ad oggi ampiamente sufficienti, comunque limitati. Un'altra ragione di questa attenzione è la crescente diffusione di servizi in tempo reale che quindi richiedono di scambiare quantità di dati considerevoli in pochissimo tempo. Un esempio di tali servizi potrebbe essere l'uso della realtà aumentata in ambito medico o di ricerca. Comprimere i dati è quindi ormai una necessità, che si farà sempre più impellente con il crescere delle dimensioni dei dati che ci ritroveremo a dover scambiare.

A partire dai primi anni novanta infatti si sono iniziate a sviluppare alcune tecniche, a cui oggi si fa riferimento come tecniche tradizionali, per comprimere le immagini. Stiamo parlando ed esempio di JPEG e del suo successore JPEG2000, di più recente sviluppo sono invece i codec BPG e VVC. Più recentemente l'attenzione dei ricercatori si è spostata su metodi che fanno uso dell'intelligenza artificiale. Questi metodi presentano diversi vantaggi rispetto ai metodi tradizionali, infatti molte volte permettono di ottenere performance migliori rispetto ai metodi classici.

In questo documento ci proponiamo di fornire una panoramica dei metodi di compressione tradizionali più usati, e di quelli che fanno uso dell'intelligenza artificiale che hanno fornito un maggiore contributo allo sviluppo di questi ultimi. Dopo aver presentato le varie tecniche vogliamo fornire una valutazione delle prestazioni in modo da poterli comparare ed evidenziare potenzialità e difetti di ognuno.

Iniziamo fornendo una spiegazione di cosa sia la compressione. La compressione è un processo che mira a minimizzare il numero di bit utilizzati per rappresentare una certa informazione senza intaccarne drasticamente la qualità. Questo obiettivo viene raggiunto riducendo le ridondanze e eliminando i dati irrilevanti. Tutti i framework di compressione consistono di una coppia codificatore-decodificatore. Data un'immagine da comprimere x il codificatore, composto da una trasformata ϵ e una funzione di quantizzazione Q , può essere espresso con la seguente formulazione

$$y = Q(\epsilon(x; \theta_\epsilon)) \quad (1)$$

Dove θ_ϵ denota i parametri del codificatore.

Per riottenere la rappresentazione dell'immagine il decodificatore ricostruisce l'immagine \hat{x} dal codice y nel seguente modo.

$$\hat{y} = D(y; \theta_D) = D(Q(\epsilon(x; \theta_\epsilon)); \theta_D) \quad (2)$$

Dove θ_D denota i parametri del decodificatore e D è una funzione che tenta di ricostruire l'immagine originale.[1]

Tutti gli algoritmi di compressione possono essere raggruppati in due macro categorie. Gli algoritmi lossy o con perdita e quelli lossless o senza perdita. Come si può già intuire dal nome gli algoritmi lossless comprimono le informazioni senza scartare informazioni, si limitano quindi ad applicare delle trasformate per ottenere delle nuove rappresentazioni più efficienti. Gli algoritmi lossy invece ammettono la possibilità di scartare delle informazioni superflue che non vanno ad intaccare drasticamente la qualità percepita dall'utilizzatore, in modo da poter comprimerle ulteriormente.

Se andiamo ad osservare gli algoritmi di codifica lossy possiamo scomporli tutti in almeno tre blocchi principali.[2]

Un primo blocco si occupa di convertire l'immagine in una rappresentazione latente, tramite l'applicazione di una trasformata, in un altro dominio che permette di rappresentare l'informazione da comprimere in modo più sparso.

Successivamente si ha un blocco che si occupa di quantizzazione, ovvero di mappare i valori in ingresso in un insieme di valori finito e di dimensione più piccola rispetto a quello di ingresso.

In questo passaggio si realizza la perdita di informazione caratteristica della codifica lossy.

Un ultimo blocco si occupa di effettuare la codifica entropica, in modo da comprimere ulteriormente l'informazione mappando i simboli usati più spesso con pochi bit.

Per concludere vorremmo fornire alcuni spunti per che potrebbero favorire l'adozione di queste

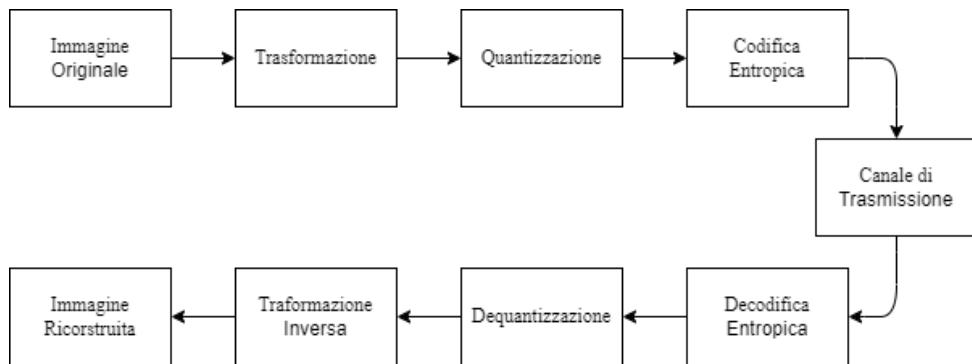


Figura 1: Diagramma di compressione Lossy

tecniche più recenti nella nostra vita digitale di tutti i giorni o suggerire nuovi possibili fronti di ricerca.

Capitolo 1

Metodi tradizionali

La caratteristica degli algoritmi di compressione tradizionali è quella di usare trasformate statiche all'interno del primo blocco, messe a punto in numerosi anni di ricerca. Questa staticità non permette a questi metodi di adattarsi dinamicamente a tutti i tipi di contenuti delle immagini. Inoltre rende il processo di sviluppo di un nuovo algoritmo di compressione un processo lungo che richiede anni di studi e progettazione. [3] Andiamo ora a parlare brevemente dei metodi che andremo a considerare quando valuteremo le prestazioni dei vari algoritmi per poterli confrontare.

1.1 JPEG

Questo metodo di compressione sviluppato nel 1992 è diventato in poco tempo, ed è attualmente il formato di compressione più diffuso. Nonostante i numerosi tentativi di sostituirlo con formati più moderni questo è rimasto ancora ad oggi il formato più usato, principalmente grazie alla sua facilità implementativa. Il JPEG si basa sull'utilizzo della DCT per realizzare la rappresentazione sparsificata dell'immagine originale. I valori prodotti dall'applicazione della trasformata vengono poi quantizzati e codificati con un metodo denominato zig-zag scan [4]

Possiamo vedere un esempio di compressione con JPEG nella figura 1.1

1.2 JPEG2000

Nel 2001 con la crescente diffusione di internet, con l'aumento di dimensione delle immagini e la richiesta di una maggiore qualità da parte degli utenti. Viene sviluppato questo nuovo formato chiamato appunto JPEG2000 per l'anno in cui è stato sviluppato.

JPEG2000 non utilizza la DCT come il suo predecessore ma viene introdotta una nuova trasformata, la DWT o Discrete Wavelet Transformat. Che si propone di meglio identificare e com-

primere i bordi delle figure che compongono le immagini, ovvero il dettaglio che ci permette di distinguere le varie regioni all'interno di un'immagine. JPEG2000 quindi voleva essere un formato di qualità superiore con una compressione più efficiente.[5]

Possiamo vedere un esempio di compressione con JPEG2000 nella figura 1.2

1.3 BPG

Questo formato è stato sviluppato da Fabrice Bellard nel 2014 come sostituto all'ormai affermato formato JPEG. Questo metodo di codifica si basa sulla codifica intra-frame del codec HEVC o H.265. [6] Bellard voleva realizzare un formato molto leggero che potesse fornire immagini più compresse rispetto a JPEG, ma con una qualità superiore.

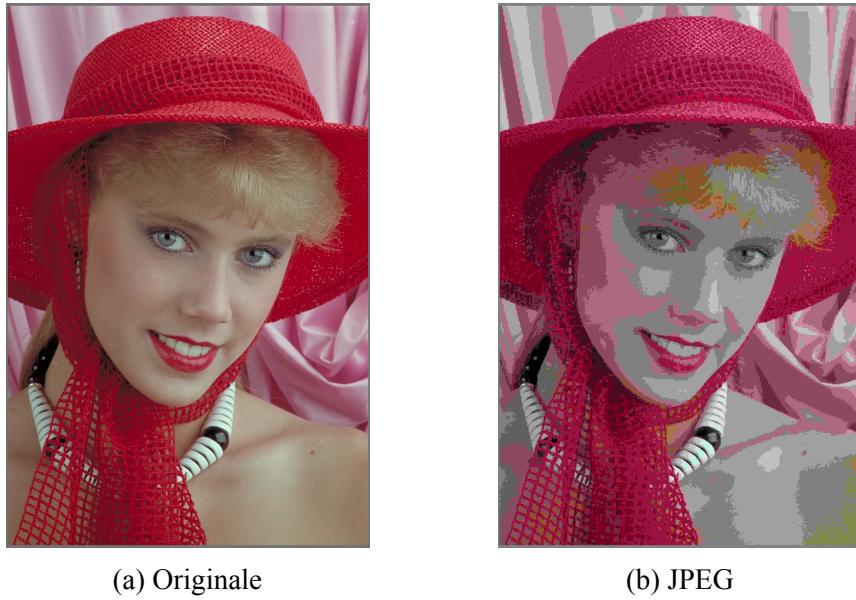
Possiamo vedere un esempio di compressione con BPG nella figura 1.3

1.4 VVC

Versatile Video Coding (VVC) o H.266 è lo standard di codifica video più recente, finalizzato nel luglio 2020. È stato sviluppato dal Joint Video Experts Team (JVET) dell'ITU-T Video Coding Experts Group (VCEG) e dell'ISO/IEC Moving Picture Experts Group (MPEG) per soddisfare la crescente richiesta di una migliore compressione video e per supportare una più ampia gamma di contenuti multimediali attuali e applicazioni emergenti come contenuti HDR, a 360°, per la Realtà Virtuale (VR) o la Realtà Aumentata (AR).[7]

Sebbene sia stato sviluppato per la compressione video, fornisce ottimi risultati anche andando a comprimere video contenenti un solo frame, ovvero delle immagini.

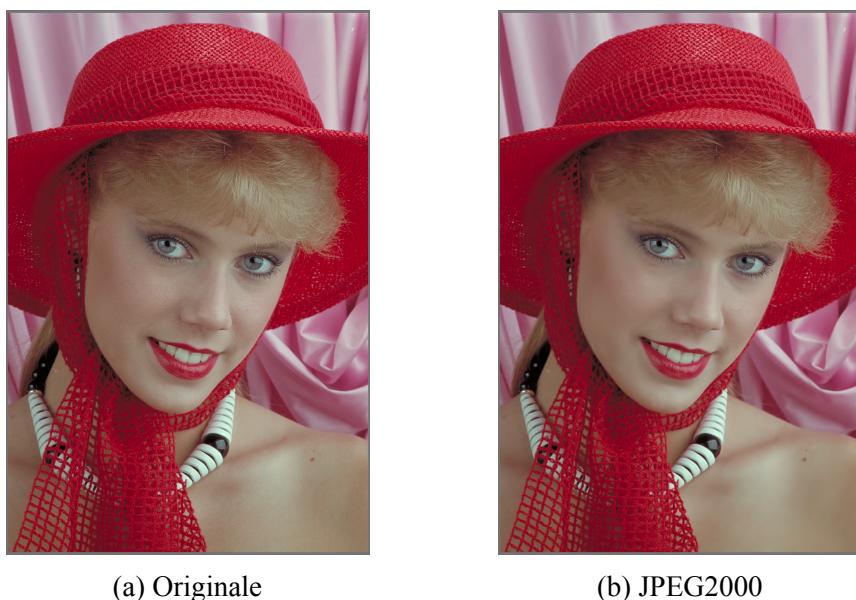
Possiamo vedere un esempio di compressione con VVC nella figura 1.4



(a) Originale

(b) JPEG

Figura 1.1: Confronto immagini PNG con JPEG bpp medio = 0.174



(a) Originale

(b) JPEG2000

Figura 1.2: Confronto immagini PNG con JPEG2000 bpp medio = 0.171



Figura 1.3: Confronto immagini BPG con JPEG bpp medio = 0.179



Figura 1.4: Confronto immagini VVC con JPEG bpp medio = 0.164

Capitolo 2

Metodi con apprendimento automatico

2.1 Ballé 2018

2.2 Cheng 2020

2.3 Yang 2021

2.4 Wang 2022

Capitolo 3

Valutazione delle prestazioni

3.1 Metriche utilizzate

3.1.1 BPP

3.1.2 Tempo di codifica

3.1.3 PSNR

3.1.4 MSSIM

3.1.5 LPIPS

3.2 Presentazione dei risultati

Capitolo 4

Sviluppi futuri

4.1 Possibili ottimizzazioni

4.1.1 Spectral ADAM

4.2 Dispositivi mobili

4.2.1 Small CAE

Capitolo 5

Conclusioni

Bibliografia

- [1] Y. Hu, W. Yang, Z. Ma e J. Liu, «Learning end-to-end lossy image compression: A benchmark,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, n. 8, pp. 4194–4211, 2021.
- [2] H. T. Sadeeq, T. H. Hameed, A. S. Abdi e A. N. Abdulfatah, «Image compression using neural networks: a review,» *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 17, n. 14, pp. 135–153, 2021.
- [3] Z. Cheng, H. Sun, M. Takeuchi e J. Katto, «Deep convolutional autoencoder-based lossy image compression,» in *2018 Picture Coding Symposium (PCS)*, IEEE, 2018, pp. 253–257.
- [4] G. Wallace, «The JPEG still picture compression standard,» *IEEE Transactions on Consumer Electronics*, vol. 38, n. 1, pp. xviii–xxxiv, 1992. doi: 10.1109/30.125072.
- [5] A. Skodras, C. Christopoulos e T. Ebrahimi, «The JPEG 2000 still image compression standard,» *IEEE Signal Processing Magazine*, vol. 18, n. 5, pp. 36–58, 2001. doi: 10.1109/79.952804.
- [6] F. Bellard, *BPG Image format*, <https://bellard.org/bpg/>, Consultato: 17-10-2023.
- [7] B. Bross, Y.-K. Wang, Y. Ye et al., «Overview of the Versatile Video Coding (VVC) Standard and its Applications,» *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, n. 10, pp. 3736–3764, 2021. doi: 10.1109/TCSVT.2021.3101953.