Comprimare si Decomprimare JPEG

Student: Alexandra-Ioana Varga, grupa 30233

Indrumator: Asist. Dr. Ing. Robert Varga

26.05.2022

# Introducere

* Descrierea problemei abordate
* Contextul problemei
* Motivare

# Studiul bibliografic

* Descrierea sumara a 2-3 metode din literatura de specialitate care ofera solutii pentru problema abordata – citarea articolelor in care s-au abordat aceste metode

# Metoda propusa

* Descrierea metodei propuse
* Detalierea conceptelor teoretice / algoritmilor folositi

# Rezultate experimentale

* Descrierea setului de date
* Testarea si validarea metodei propuse

# Concluzii

* Un sumar al realizarile obtinute,
* Gradul de implinire a obiectivelor propuse
* Observatii asupra rezultatelor obtinute
* Directii viitoare de dezvoltare

# Bibliografie

* Lista referintelor folosite

# Introducere

* **Descrierea problemei abordate**

Compresia imaginii este reducerea la minimum a dimensiunii în octeți a unui fișier grafic fără a degrada calitatea imaginii la un nivel inacceptabil. Reducerea dimensiunii fișierului permite stocarea mai multor imagini într-o anumită cantitate de disc sau spațiu de memorie. De asemenea, reduce timpul necesar pentru ca imaginile să fie trimise prin Internet sau descărcate de pe paginile Web.

* **Contextul problemei**

Obiectivul principal în compresia imaginii este:

* Stocarea datelor într-o formă eficientă;
* Transmiterea datelor într-o formă eficientă;

Compresia imaginii se clasifica in cea “**cu pierderi”** sau “**fără pierderi”**;

* **Motivare**

JPEG = Joint Photographic Experts Group. Standard ISO şi ITU-T 1982-1992.

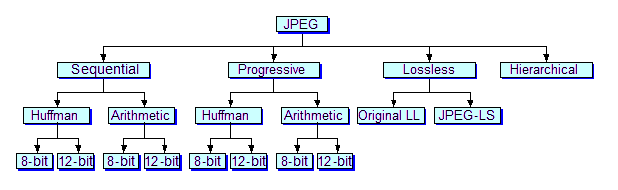
Este primul standard internațional în compresia imaginii. Este folosit pe scară largă astăzi, de aceea am dorit sa implementez o astfel de operatie asupra imaginilor in cadrul laboratorului “Procesare de Imagini”.

# Studiul bibliographic

* **Metode descrise in literatura de specialitate**

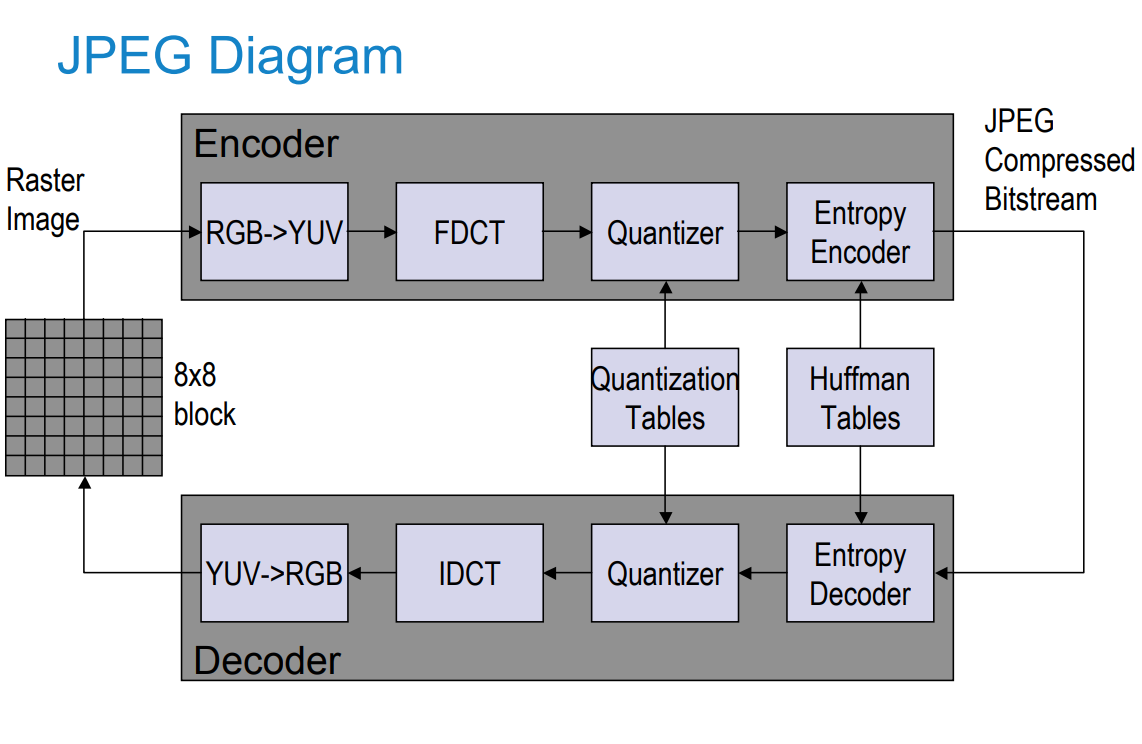
Compresia unei imagini ar putea fi implementata atât cu pierderi, cât și fără pierderi. Tehnica pe care am implementat-o este cea de compresie **cu pierderi**. Modul de **operare** asupra imaginilor se clasifica in:

* **Secvenţial**: imaginile în modul secvenţial sunt codificate de sus în jos. Modul secvenţial acceptă date mostre cu 8 şi 12 biţi de precizie. În JPEG secvenţial, fiecare componentă de culoare este complet codificată într-o singură scanare. În modul secvenţial, două procese alternative de codificare a entropiei sunt definite de standardul JPEG: unul foloseşte codarea Huffman; celălalt folosește codificarea aritmetică.
* **Progresiv**: În imaginile JPEG progresive, componentele sunt codificate în mai multe scanări. Datele comprimate pentru fiecare componentă sunt plasate în minim 2 și până la 896 de scanări. Scanările inițiale creează o versiune brută a imaginii, în timp ce scanările ulterioare o rafinează.
* **Fara pierderi**: păstrează imaginea exactă, originală, raport mic de compresie, utilizare mai mică
* **Ierarhic**: JPEG este un mod super-progresiv în care imaginea este împărțită într-un număr de subimagini numite cadre. Un cadru este o colecție de una sau mai multe scanări. În modul ierarhic, primul cadru creează o versiune de imagine cu rezoluție scăzută. Cadrele rămase rafinează imaginea prin creșterea soluției.

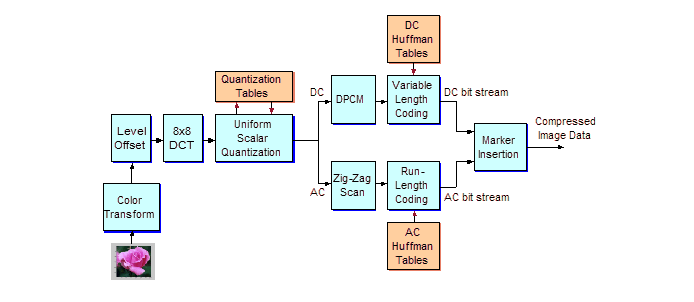


# Metoda propusa

* **Descrierea metodei propuse**

****

* **Detalierea conceptelor teoretice / algoritmilor folositi**
* Imaginea de intrare este împărțită într-un bloc mic care are dimensiuni de 8x8. Această dimensiune este însumată până la 64 de unități. Fiecare unitate a imaginii se numește pixel.
* JPEG folosește modelul [Y,Cb,Cr] în loc să folosească modelul [R,G,B]. Deci, în al 2-lea pas, RGB este convertit în YCbCr.
* După conversia culorilor, acesta este redirecționat către DCT. DCT folosește o funcție cosinus și nu folosește numere complexe. Acesta convertește informațiile care se află într-un bloc de pixeli din domeniul spațial în domeniul frecvenței.
* Oamenii nu pot vedea aspecte importante ale imaginii, deoarece au frecvențe înalte. Matricea după conversia DCT poate păstra doar valorile la frecvența cea mai joasă până la un anumit punct. Cuantizarea este utilizată pentru a reduce numărul de biți per probă.
* Scanarea în zig-zag este utilizată pentru a mapa matricea 8x8 la un vector 1x64. Scanarea în zig-zag este utilizată pentru a grupa coeficienții de frecvență joasă la nivelul superior al vectorului și coeficientul ridicat în partea de jos. Pentru a elimina numărul mare de zero-uri din matricea cuantificată, se utilizează matricea în zig-zag.
* Următorul pas este vectorizarea, modulația de cod de impuls diferit (DPCM) este aplicată componentei DC. Componentele DC sunt mari și variază, dar sunt de obicei apropiate de valoarea anterioară. DPCM codifică diferența dintre blocul curent și blocul anterior.
* În acest pas, codificarea lungimii de rulare (RLE) este aplicată componentelor AC. Acest lucru se face deoarece componentele AC au o mulțime de zerouri în ele. Acesta codifică în perechi de (sărire, valoare) în care skip este o valoare diferită de zero și valoarea este valoarea codificată reală a componentelor non zero.
* **Metoda inversa** a compresiei urmareste pasii explicati anterior in ordine inversa utilizand fuctia IDCT aplicata pentru a putea decodifica blocurile imaginii comprimate.



# Rezultate experimentale

* **Descrierea setului de date**

Setul de date consta in 10 imagini .bmp asupra carora se vor aplica algoritmii de comprimare si decomprimare.

* **Testarea si validarea metodei propuse**

Testarea si validarea metodei propuse consta in:

* Se aplica algoritmul de comprimare pe rand asupra imaginilor .bmp;
* Se aplica algoritmul de decomprimare pe rand asupra imaginilor comprimate, stocate in fisier binar sub forma de bloc-uri de compresie;
* Daca dimensiunea initiala a imaginilor originale necomprimate este aproximativ egala (marja de eroare minima) cu dimensiunea imaginilor decomprimate cu algoritmul implementat, putem concluziona ca algoritmul a fost implementat corect si corespunzator;

# Concluzii

* **Un sumar al realizarile obtinute (Imagine .bmp originala vs. imagine .jpg decomprimata)**

**O imagine care conține text, pisică, așezat, ecran

Descriere generată automat**

**O imagine care conține text, roșu, sport

Descriere generată automat**

**O imagine care conține text, exterior, semn

Descriere generată automat**

* **Gradul de implinire a obiectivelor propuse**

Conform algoritmului implementat anterior am parcurs etapele de prelucrare a imaginii pas cu pas.

Stocarea imaginii comprimate (blocurile de compresie) a fost facuta intr-un fisier .txt si de asemenea intr-un fisier .jpg. Pe baza fisierului ce contine imaginea comprimata .jpg am implementat decompresia imaginii, rezultatele finale fiind exemplificate in cele de mai sus. Pentru operatia de decompresie se reia algoritmul in ordine inversa si aplicata functia idct pentru decodificarea blocurilor comprimate ale imaginii).

Obiective neimplinite ale proiectului sunt:

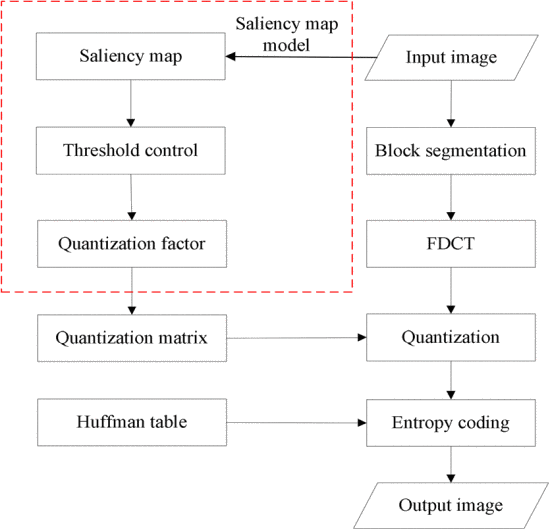
* Funtionalitate zig-zag + RLE in cadrul algoritmului, aceasta functie este implementata dar neapelata);
* Functionalitate dct/idct (functii implementate si neapelate, sunt folosite functiile implementate din OpenCV);
* Scriere in fisier binar si decodificare secvente binare (am folosit fisier .txt).
* **Observatii asupra rezultatelor obtinute**

Rezultatele algoritmului de compresie aplicat consta **in micsorarea dimensiunii** stocate pe disc a imaginilor de test. Aceasta modificare a dimensiunii este vizibila (variatii de dimensiuni noi de la **MB -> KB**). Aceasta operatie de compresie are ca consecinta **pierderea din calitatea culorilor** usor vizibila ochiului uman.

* **Directii viitoare de dezvoltare**

Aplicarea in totalitate a obiectivele neatinse de mine pot reprezenta un prim pas in dezvoltarea acestui algoritm.

Utilizare **harta de proeminenta**:



În mecanismul de atenție vizuală, hărțile de proeminență extrase din imaginile de intrare sunt utilizate în mod obișnuit pentru a reflecta caracteristicile importante ale imaginilor. Algoritmul descompune procesul de extragere a hărții de proeminență în trei părți:

* extragerea caracteristicilor;
* calculul proeminenței;
* normalizarea și combinarea.

# Bibliografie

* **Lista referintelor folosite**

Surse informatice utilizate:

<http://pi.math.cornell.edu/~web6140/TopTenAlgorithms/JPEG.html>

<https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/dct.htm>

<http://www0.cs.ucl.ac.uk/teaching/GZ05/07-images.pdf>

<https://www.tutorialspoint.com/dip/Introduction_to_JPEG_compression.htm>

<https://www.javatpoint.com/jpeg-compression>

<https://www.ece.ucdavis.edu/cerl/reliablejpeg/compression/>

<https://www.geeksforgeeks.org/process-of-jpeg-data-compression/>

<https://ieeexplore.ieee.org/document/6469992>

<https://www.geeksforgeeks.org/discrete-cosine-transform-algorithm-program/>