

SACCDMM - Curs 12

Procesarea imaginilor JPEG direct in domeniul comprimat

Sl.Dr.Ing. Camelia FLOREA

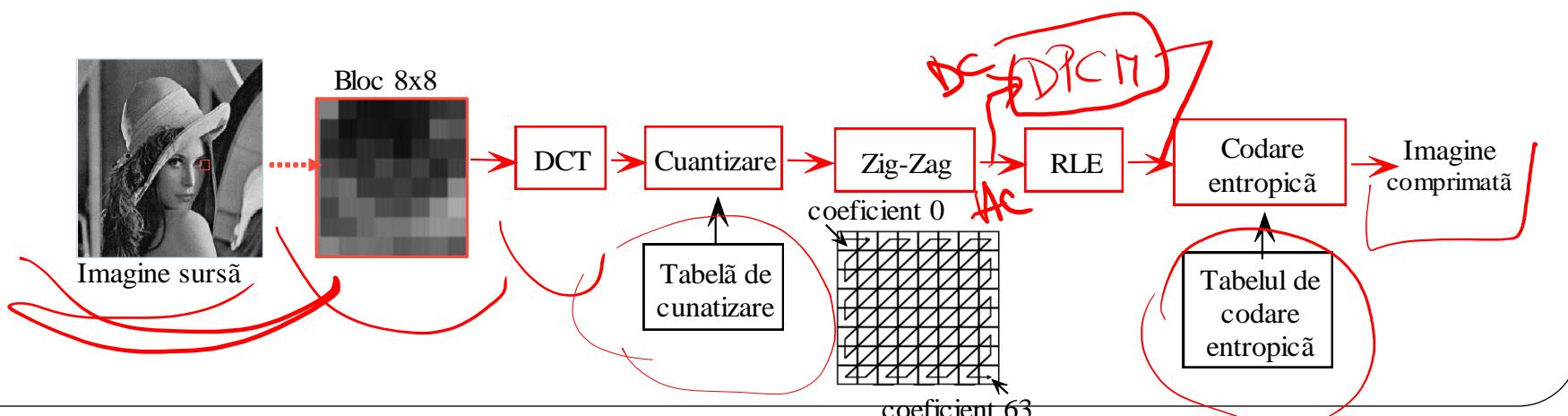
Introducere

- Odată cu **dezvoltarea exponențială a tehnologiilor care implică informație digitală**, pornind de la simplul format text și ajungând la imagini digitale de înaltă rezoluție, apare și necesitatea manevrării într-un mod optim a acestora. Astfel, problemele deschise de necesitatea **transmiterii** imaginilor la distanță, **stocarea** lor pe compact discuri cu păstrarea calității la o dimensiune binară minimă, **integrarea** datelor în diferite contexte și medii, precum și **procesarea acestora în timp real**, trebuie abordate cu atenție.
- Trebuie menționată în acest context **eficiența formatului JPEG**, un format utilizat pe scară largă pentru stocarea digitală a imaginilor. Pe lângă o rată înaltă de compresie, apare necesitatea de păstrare a calității informațiilor originale, formatul JPEG îmbinând cele două deziderate.
- Astfel, devine naturală tendința de prelucrare a imaginilor JPEG direct în domeniul comprimat. Operațiile de prelucrare a imaginilor JPEG în domeniul comprimat pot opera adesea mult mai repede decât corespondentele din domeniul spațial fapt dat de evitarea decompresia/recompresia înainte/după prelucrare și de faptul că în domeniul comprimat se procesează mai puține date. Rămân însă câteva probleme de rezolvat: nu există tehnici adecvate în domeniul comprimat de efectuare a tuturor operațiilor neliniare.

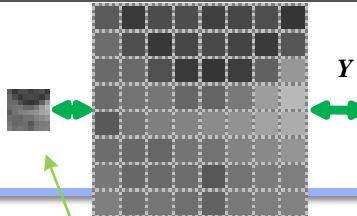
- În ultima perioadă, a existat un interes deosebit în dezvoltarea unor tehnici de căutare a imaginilor după conținutul lor în baze de date, pe web, etc...
- Librariile digitale manipulează foarte multe imagini – datorită limitelor date de spațiul de stocare și timpului necesar manipулării lor – informația este reprezentată în format comprimat => tehniciile utilizate pentru segmentarea, clasificarea sau indexarea imaginilor direct în domeniul comprimat au devenit un subiect important în domeniul librariilor digitale.
- Dintre formatele de compresie, formatul JPEG a fost utilizat mai mult decât celelalte (mai mult de 95 % din imaginile disponibile pe internet sunt stocate ca JPEG-uri)
- Chiar și majoritatea imaginilor medicale sunt stocate în formatul JPEG, datorită avantajelor oferite de acest format:
 - Spațiul redus necesar pentru ~~stocarea~~ imaginilor, și
 - Performanțe superioare la transmiterea imaginilor (via internet).

Compresia datelor utilizând formatul JPEG

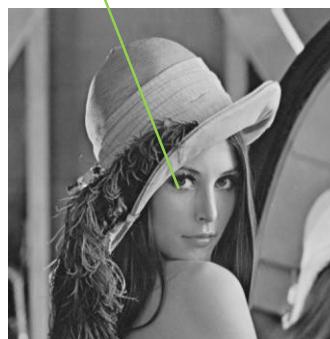
- Spatiul de culoare (RGB \rightarrow YUV) $YCbCr$
- Imaginea este impartita in blocuri de 8x8 pixeli
- Valorile sunt scalate simetric fata de 0 (din [0, 255] in [-128, 127]) $\leftarrow 128$
- Fiecare bloc de 8x8 pixeli este procesat pentru compresie
 - DCT este aplicata pe fiecare bloc \Rightarrow se obtin coeficientii DCT (DC si AC)
 - Coeficientii DCT sunt cuantizati – coeficientii care au valori mici sunt cuantizati la zero
 - Ordonarea in zig-zag a blocului DCT
 - RLE (Run Length Encoding)
 - Codarea entropica



Extragerea unui bloc de 8×8 pixeli din componenta Y și aplicarea pașilor pentru compresie în format JPEG.



Bloc de 8×8 pixeli extras din componenta de luminanță



Componenta de luminanță a imaginii

4	3	2	4	6	10	13	16
3	3	3	5	7	14	15	14
2	3	4	6	10	14	17	14
4	5	6	9	13	22	20	16
6	7	10	13	17	28	26	19
10	14	14	22	28	26	28	23
13	15	17	20	26	28	30	26
16	14	14	16	19	23	26	25

Tabela de cuantizare

00	01	02	03	04	05	06	07
08	09	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Scanare în Zig-Zag

90	57	76	77	64	71	78	53
109	79	56	70	73	68	57	85
106	105	80	58	53	62	97	151
107	115	118	102	100	120	157	184
89	115	127	131	138	146	162	171
109	104	102	116	114	131	137	149
119	103	105	108	87	115	124	125
127	111	118	110	98	120	125	129

Y_S

-38	-71	-52	-51	-64	-57	-50	-75
-19	-49	-72	-58	-55	-60	-71	-43
-22	-23	-48	-70	-75	-66	-31	23
-21	-13	-10	-26	-28	-8	29	56
-39	-13	-1	3	10	18	34	43
-19	-24	-26	-12	-14	3	9	21
-9	-25	-23	-20	-41	-13	-4	-3
-1	-17	-10	-18	-30	-8	-3	1

Y_{DCT}

-181.12	-58.32	73.49	-18.54	7.62	11.75	9.20	-7.35
-134.15	30.04	12.30	3.86	15.69	-5.92	-2.77	6.69
-107.32	70.10	-19.33	33.69	6.99	23.20	10.06	-1.74
6.09	-11.82	-55.59	21.18	-3.29	15.1	-2.20	-0.31
53.62	-23.28	-27.78	-7.03	-19.12	11.52	0.54	10.92
-6.22	-2.55	20.13	-10.91	-12.62	15.78	1.67	2.66
-7.90	-3.07	15.56	-9.11	-3.03	-0.86	-0.41	3.14
8.01	0.18	-0.99	-11.05	-7.15	-0.77	-2.87	1.97

Y_Q

-45	-19	37	-5	1	1	1	0
-45	10	4	1	2	0	0	0
-54	23	-5	6	1	2	1	0
2	-2	-9	2	0	1	0	0
9	-3	-3	-1	-1	0	0	1
-1	0	1	0	0	1	0	0
-1	0	1	0	0	0	0	0
-1	0	0	-1	0	0	0	0

Y_{ZZ}

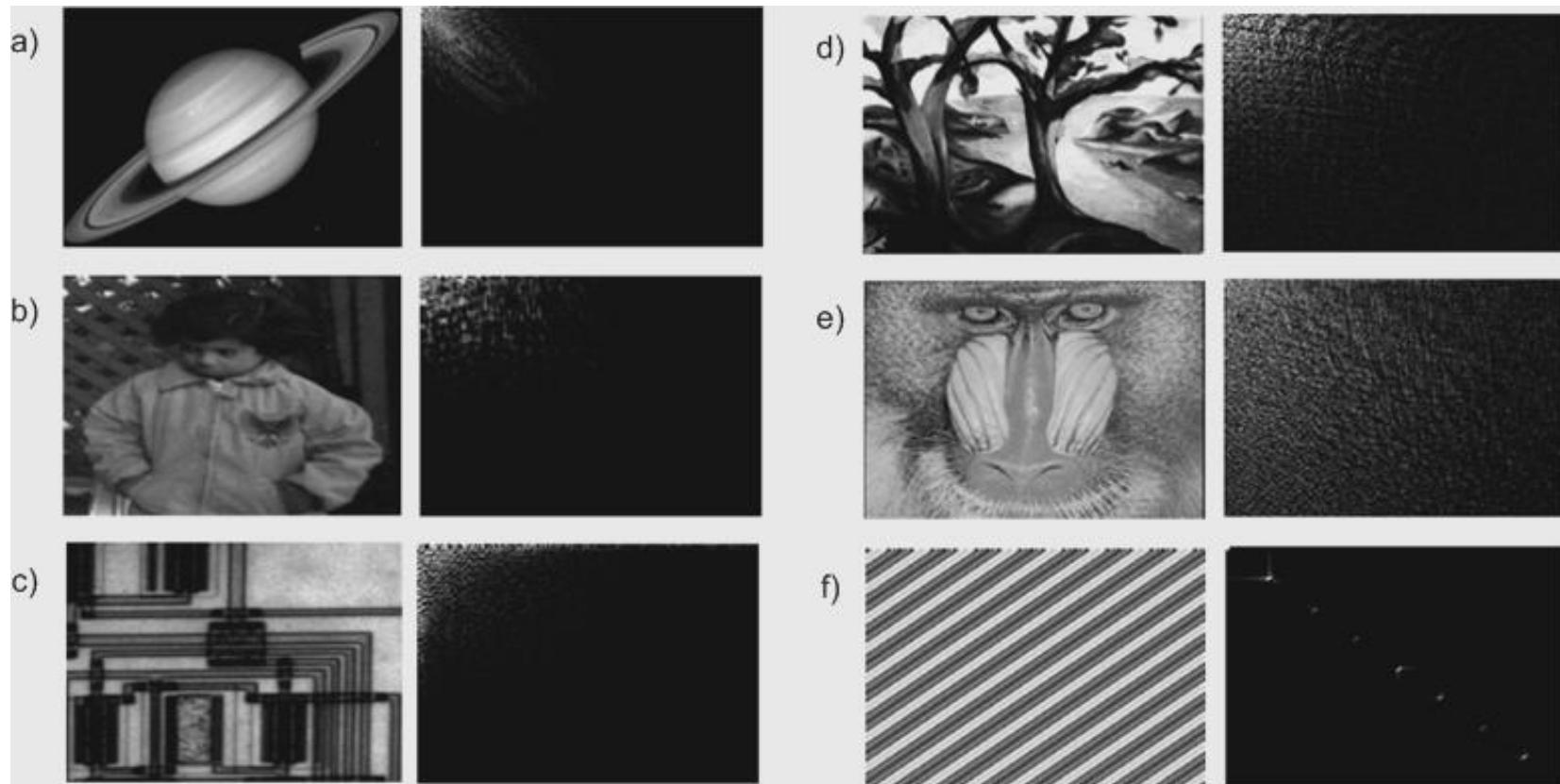
[-45 -19 -45 -54 10 37 -5 4 23 2 9 -2 -5 1 1 1 2 6 -9 -3 -1 -1 0 -3 2 1
0 1 0 0 2 0 -1 1 0 -1 0 1 0 -1 1 1 0 0 0 0 0 0 0 -1 0 1 0 0 1 0 0 0 0 0
0 0 0 0]

Y_{RLE}

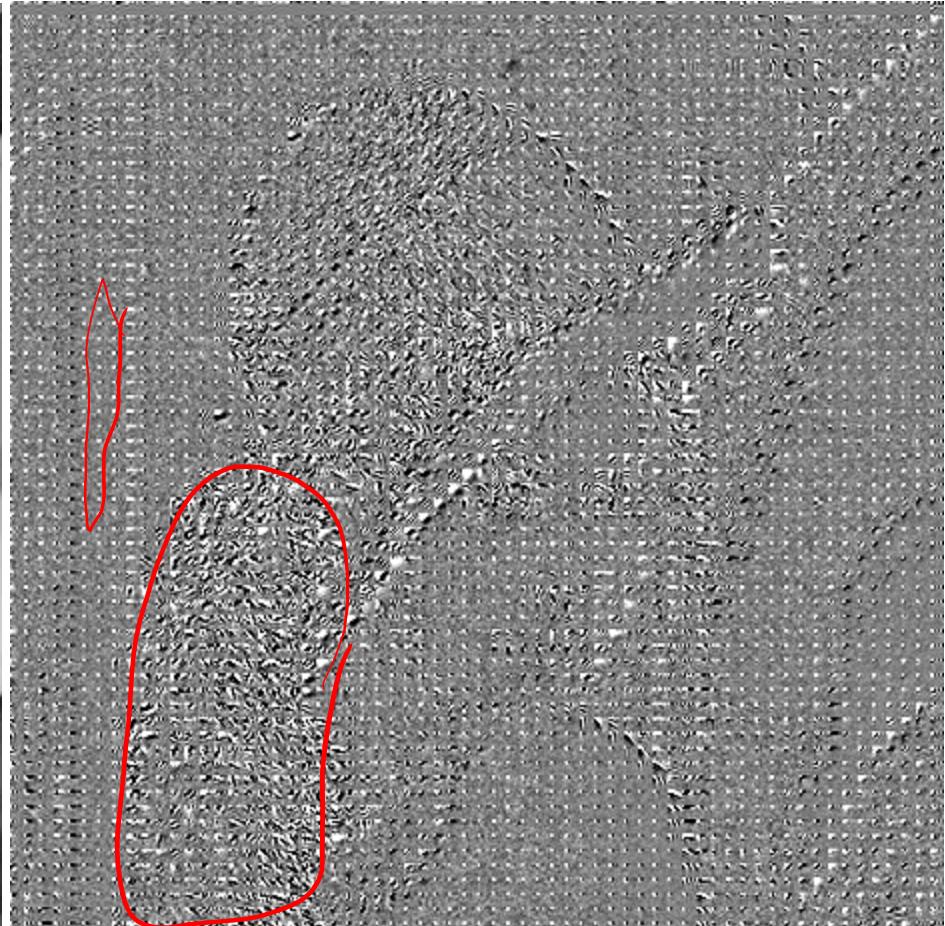
[-45 0 -19 0 -45 0 -54 0 10 0 37 0 -5 0 4 0 23 0 2 0 9 0 -2 0 -5 0 1 0
1 0 1 0 2 0 6 0 -9 -3 0 -1 0 -1 1 -3 0 2 0 1 1 1 2 2 1 -1 0 1 1 -1 1 1
1 -1 0 1 0 1 7 -1 1 1 2 1 0 0]

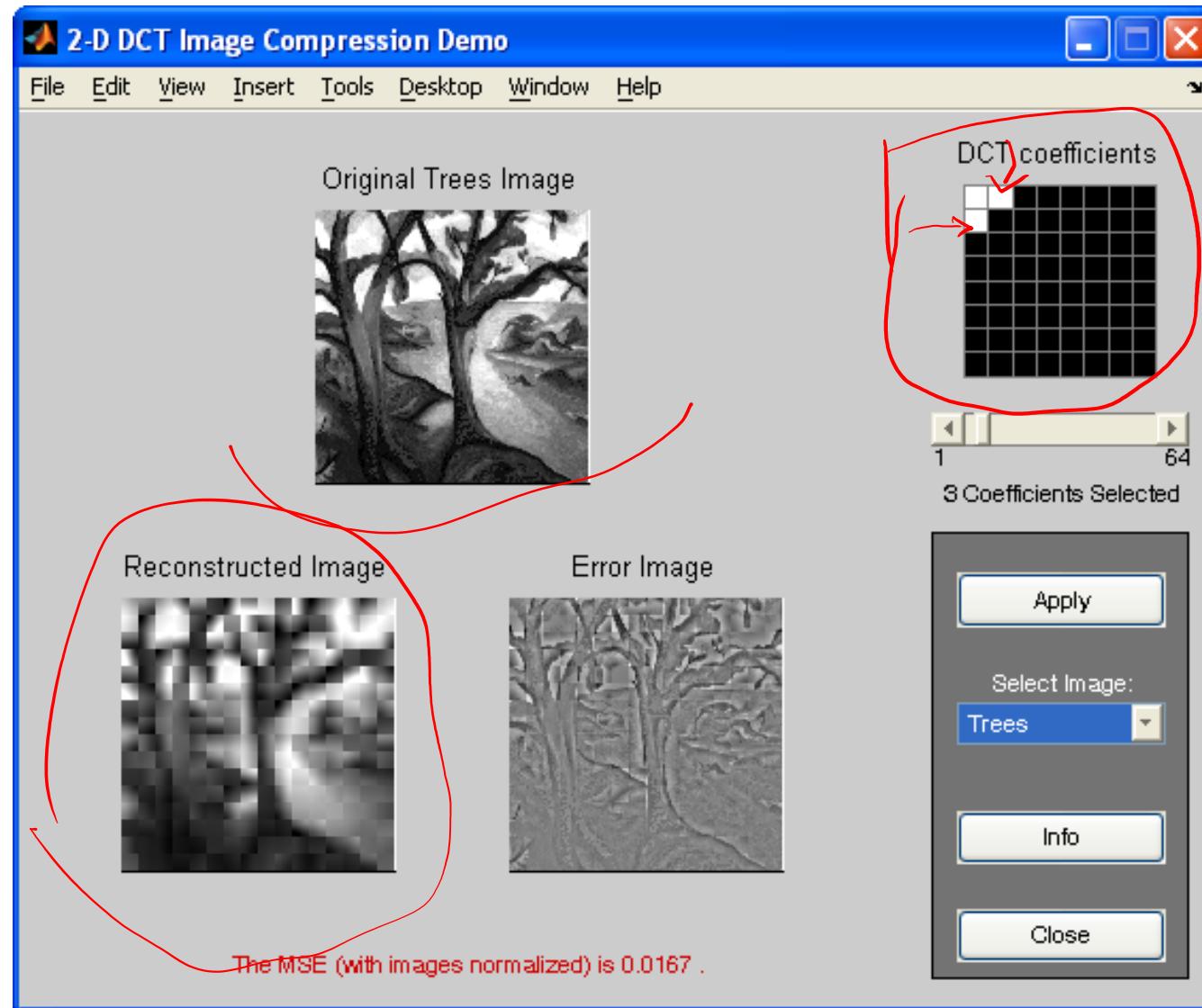
Coeficienții DCT

- În coeficienții DCT este prezentă toată informația din blocul spațial de imagine, într-o altă formă de reprezentare:
 - Coeficientul DC – luminanță/crominanță medie dintr-un bloc de pixeli
 - Coeficienții AC – modul de variație a luminanței/crominanței din acel bloc (față de coeficientul DC)



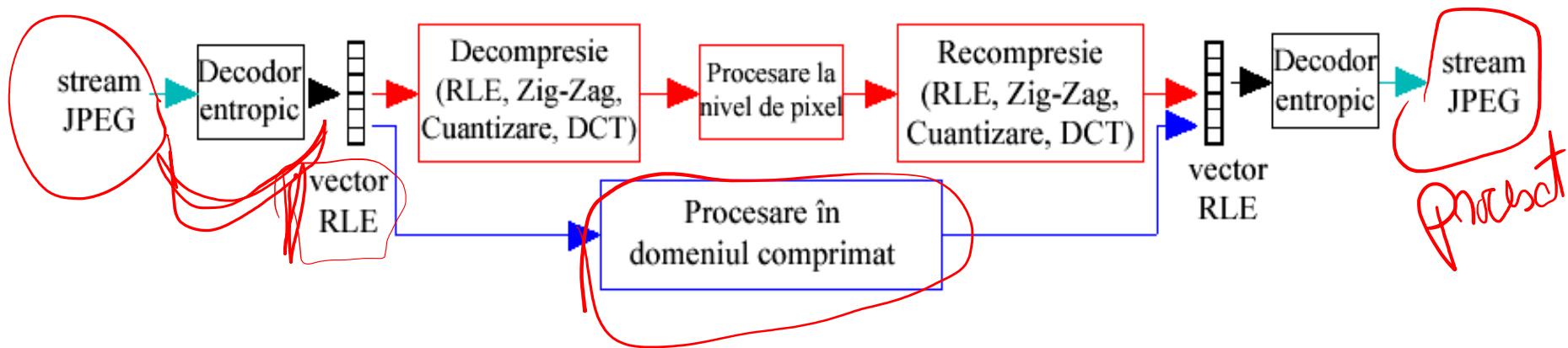
Imaginea in domeniul comprimat JPEG





Procesarea imaginilor comprimate JPEG

- Există două moduri de procesare:



- Avand imaginea stocata in formatul comprimat JPEG este mult mai eficient sa fie procesata direct, fara:
 - realizarea decompresiei, procesarea la nivel de pixel, si recompresia.
- Procesarea in domeniul comprimat este realizata pe vectorul RLE.
- Vectorul RLE contine informatie despre valoarea medie si modul de variatie a luminantei/ crominantei din blocul de 8x8

- Reprezentarea imaginii în domeniul transformatei cosinus discrete (DCT) este mai eficientă din perspectiva:
 - decorelării datelor comparativ cu reprezentarea spațială
 - compactării energiei totale a imaginii într-un număr relativ mic de coeficienți;
- Coeficienții DCT din blocuri de 8×8 pixeli disponibili în formatul JPEG furnizează direct informații de culoare și textură locală, în formă compactă, fără transformări suplimentare
 - => putem extrage ușor discriminatori de textura (ex. energia de curent alternativ (AC) din fiecare bloc);
 - => putem clasifica blocurile de pixeli în domeniul comprimat \Leftrightarrow segmentare a imaginilor JPEG în domeniul comprimat

Operații liniare de prelucrare a imaginilor în domeniul comprimat

- Prelucrarea imaginilor în domeniul comprimat \Leftrightarrow transpunerea algoritmilor de prelucrare spatială a pixelilor în domeniul comprimat;
- Transpunerea algoritmilor liniari în domeniul comprimat:
 - foarte simplă – datorită liniarității DCT
 - => algoritmi mult mai rapizi prin:
 - evitarea decompresiei și recompresiei;
 - reducerea numărului de date de prelucrat fata de domeniul spațial (după cuantizare majoritatea coeficienților AC sunt zero)
 - => categoria principală de operații: “*aritmetică pixelilor*”: adunare cu o constantă, înmulțire cu o constantă, adunarea pixelilor, înmulțirea pixelilor

Operații neliniare de prelucrare a imaginilor în domeniul comprimat

- Transpunerea algoritmilor neliniari în domeniul comprimat:
 - nu este directă; unii algoritmi sunt dificil de reformulat;
 - în ipoteza determinării relației corecte pentru implementare în domeniul comprimat, ofera avantajul unei complexitati numerice reduse
 - reprezinta o tendinta actuala in domeniu

Algoritmii liniari de prelucrare a imaginilor în domeniul comprimat

- Aritmetica pixelilor
 - adunare scalară (se adună cu o constată valoarea luminanței din fiecare pixel);
 - înmulțire scalară (înmulțește valoarea luminanței din fiecare pixel din imagine cu o constată);
 - adunare a pixelilor (adună valorile luminanțelor pixelilor din două imagini);
 - înmulțire a pixelilor (înmulțește valorile luminanțelor pixelilor din două imagini);

$$h[i, j] = \alpha \cdot f[i, j] \Leftrightarrow H_Q[u, v] = \frac{\alpha \cdot q_F(u, v)}{q_H(u, v)} \cdot F_Q[u, v].$$

$\mathcal{F} = \text{DCT}(f)$

Coef. DC

	Spatial domain signal - x	Transform domain signal - X
Scalar addition	$[f] + \alpha$	$[F] + \begin{bmatrix} 8\alpha/Q_{00} & 0 \\ 0 & 0 \end{bmatrix}$
Scalar Multiplication	$\alpha[f]$	$\alpha[F]$
Pixel Addition	$[f] + [g]$	$[F] + [G]$
Pixel Multiplication	$[f] \bullet [g]$	$[F] \otimes [G]$

Aplicații a aritmeticii pixelilor

- Negativare

Componenta Y $\rightarrow f$

$$g(x, y) = 256 - f(x, y) \mid -128$$

$$g(x, y) - 128 = 256 - f(x, y) - 128$$

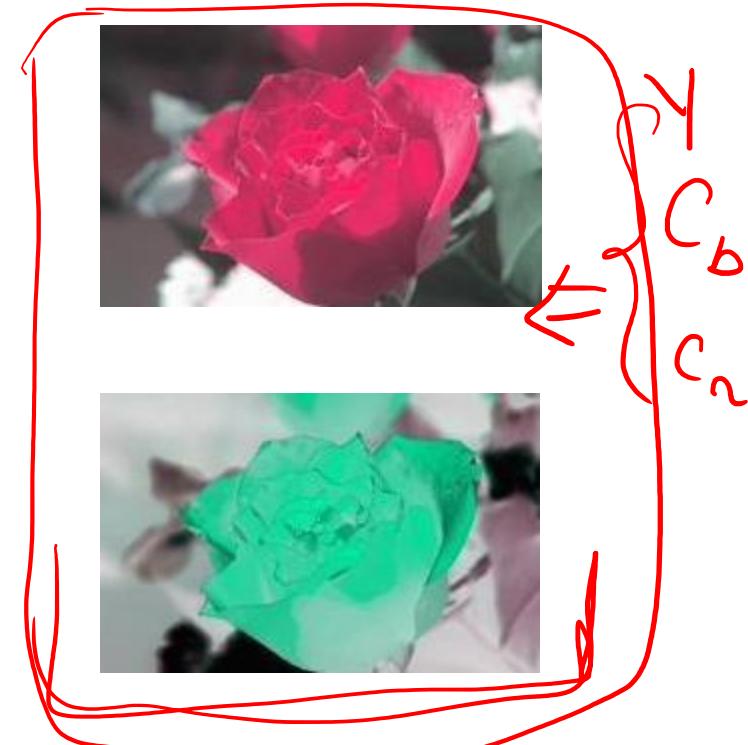
$$g(x, y) - 128 = 128 - f(x, y)$$

$$g(x, y) - 128 = -(f(x, y) - 128)$$

$$DCT(g - 128) = -DCT(f - 128)$$

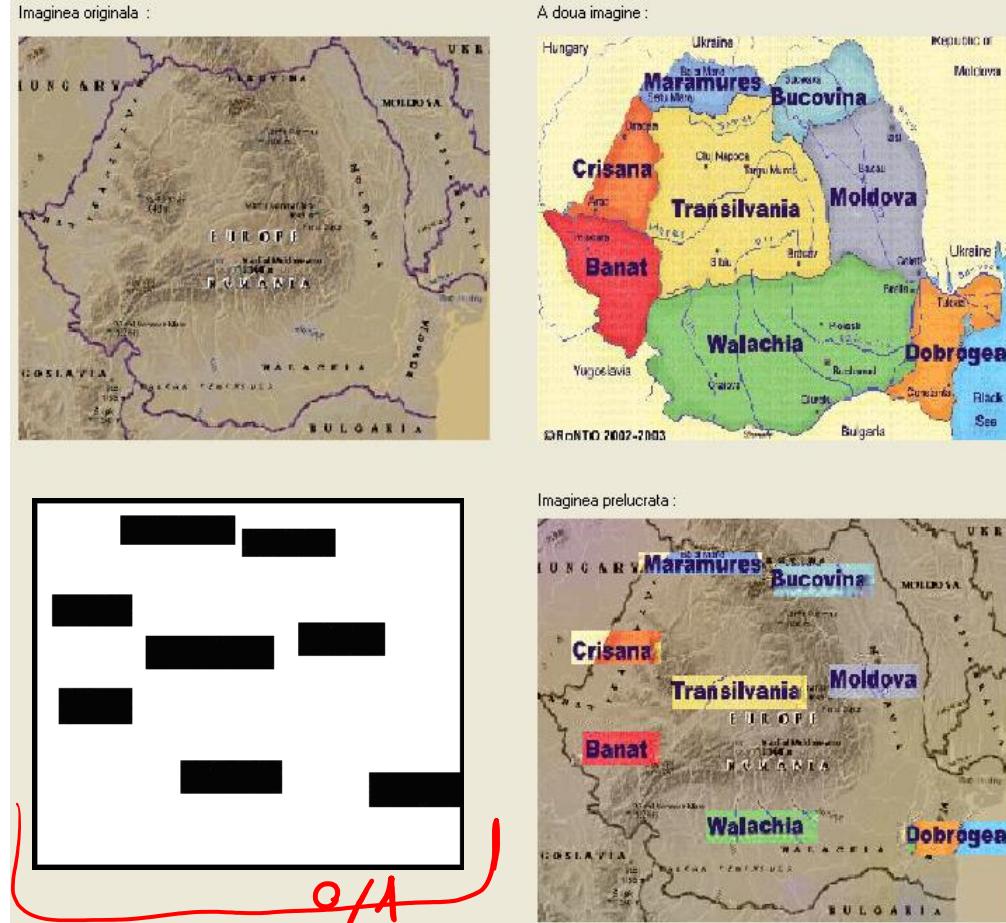
$$\text{Val}_{DCT_{pq}} = -\text{Val}_{DCT_{1pq}}$$

Pas 1 - axene valori tezoare
 $L = 128$



Aplicații a aritmeticii pixelilor

- Compunere de imagini
(ex. suprapunerea buletinului meteo pe hartă)



$$g(x, y) = m(x, y) \cdot f(x, y) + (1 - m(x, y)) \cdot b(x, y)$$

$$= \frac{b}{f}$$

Obturare progresivă

$$g(x, y) = \alpha f_2(x, y) + (1 - \alpha) f_1(x, y)$$

$\underbrace{\alpha f_2(x, y)}_{=0} + \underbrace{(1 - \alpha) f_1(x, y)}_{=0}$



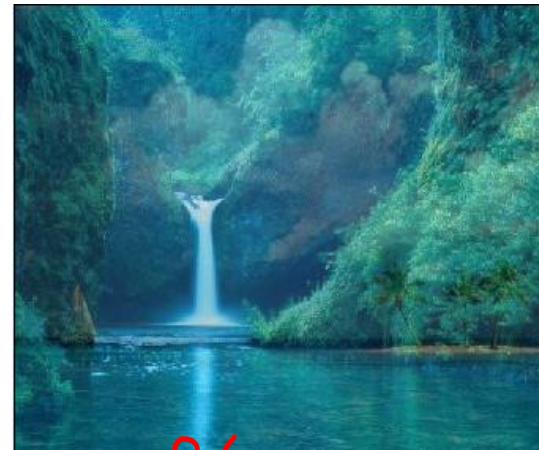
$\alpha=1$



~~$0.8 f_2 + 0.2 f_1$~~



0.1α



0.16



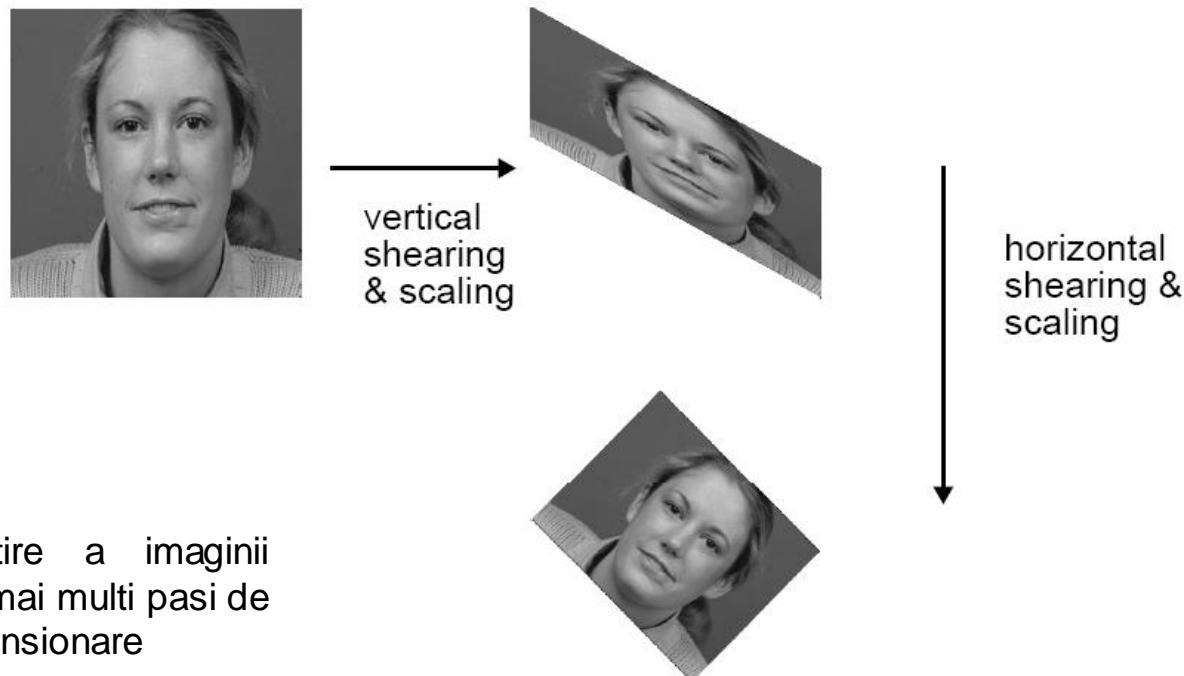
0.2



$\alpha=0 \Rightarrow f_1$

Transformari Geometrice

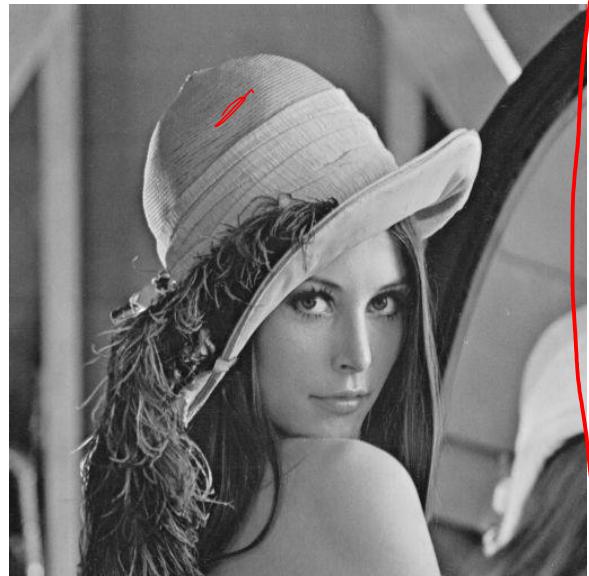
- Tehnici pentru manipularea imaginilor in domeniu comprimat pentru efecte speciale, corectia deformarilor: redimensionarea imaginilor, translatia, rotatia, warping (deformarea).
- Functioneaza prin simpla rearanjare a datelor in domeniu comprimat (coeficientilor DCT), fara a se decomprima intreaga imagine.
- Aceste algoritmi sunt aplicabili pe orice transformata ortogonală cum ar fi: DCT, DFT, ...



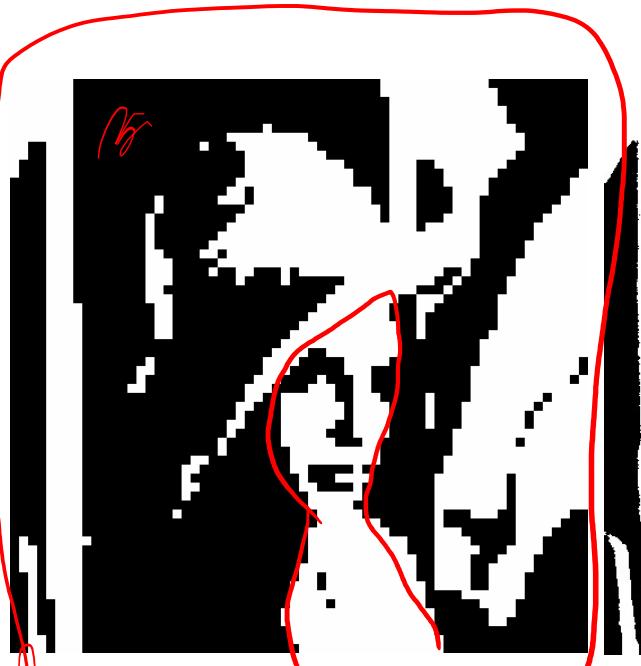
Implementarea operației neliniare de tip comparare cu prag în domeniul comprimat

- Problema adresată
 - Implementarea direct în domeniul comprimat a unor algoritmi care necesită una sau mai multe comparații cu prag a luminanței fiecărui pixel
 - Coeficienții DCT nu furnizează direct luminanțele pixelilor decât în cazul blocurilor perfect uniforme => pentru implementarea comparației cu prag a fiecarei luminanțe ar fi necesară decompresia
- Abordare
 - Realizarea unui algoritm adaptiv
 - blocurile cu luminanță uniformă se procesează direct în domeniul comprimat
 - dar blocurile care concentrează o cantitate mare de energie de curent alternativ (blocuri neuniforme) vor fi decomprime și se procesate individual - fiecare luminantă.
- Avantaje
 - Decompresia nu mai este necesară pentru toată imaginea
 - Pentru blocurile procesate în domeniul comprimat comparația cu pragul se face o singură dată pentru tot blocul

Binarizarea unei imaginii



Imaginea originală
“Lena.jpg”;



Imaginea procesată în
domeniul comprimat;



Imaginea procesată
la nivel de pixel

b7 componenti:

→ 1 componentă
a coef DC

Utilizarea algoritmului cu decompresie adaptivă

- Pentru fiecare bloc de 8×8 pixeli din imaginea digitală comprimată JPEG:

- Calculează energia medie a coeficientilor AC din blocul DCT, E_{AC} .

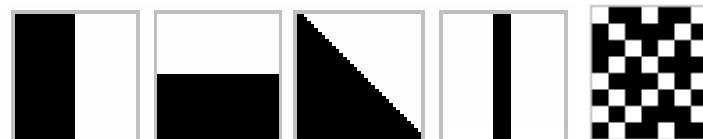
$$E_{AC} = \frac{\sum_{i=1}^{(N_{RLE}-1)/2} (u_{RLE}[2i])^2}{(N_{RLE}-1)/2}$$

- Dacă $E_{AC} < e_{\text{thd}}$

=> blocul este suficient de uniform pentru a fi procesat direct în domeniul comprimat.

Dacă $E_{AC} \geq e_{\text{thd}}$

=> blocul are un număr semnificativ de detalii => se va decomprima; fiecare pixel din bloc este comparat individual cu pragul de luminanta.



- Valoarea e_{thd} = valoarea parametrului de selecție a blocurilor uniforme fata de blocurile cu număr semnificativ de detalii.

Selectia parametrului e_{thd}

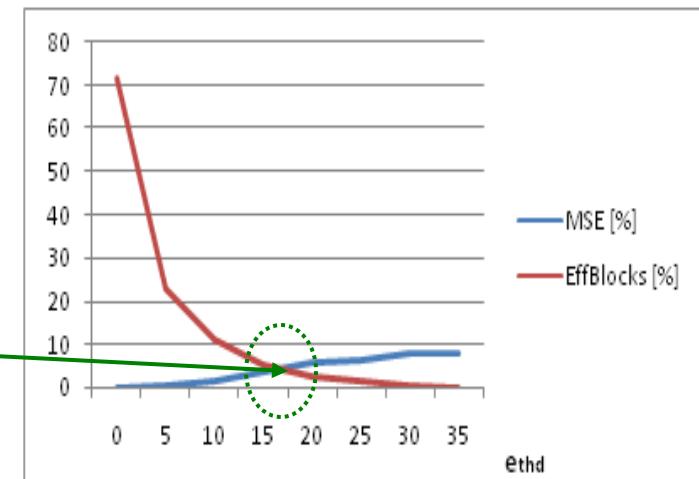
- Pentru o clasă de imagini dată, valoarea e_{thd} optimă = valoarea care oferă cel mai bun compromis între:
 - complexitatea de calcul
 - și
 - calitatea imaginii procesate.

Determinarea e_{thd} :

- experimental, prin examinarea variatiilor
 - graficului erorii medii pătrate (MSE - Mean Square Error)
 - și
 - graficului EffBlocks = procentul numărului de blocuri decomprime din numărul total de blocuri din imagine (la aplicarea algoritmului propus),

în funcție de valorile parametrului e_{thd}

Regiunea
corespunzătoare
valorii optime a
parametrului e_{thd}



Binarizarea imaginilor în domeniul comprimat folosind algoritmul adaptiv

- Formularea algoritmului adaptiv de binarizare a imaginilor comprimate JPEG propus:
 1. Alegerea pragului T de binarizare (implicit 128) => DC_{thd} (implicit 0).
 2. Calculul conținutului mediu de energie E_{AC} pentru fiecare bloc de 8×8 pixeli din imaginea comprimata JPEG
 - Dacă $E_{AC} < e_{thd}$, atunci blocul se poate considera de luminanță uniformă și se binarizează direct în domeniul comprimat:

$$u_{RLE}[0] = \begin{cases} -127, & \text{dacă } u_{RLE}[0] < DC_{thd} \\ 127, & \text{dacă } u_{RLE}[0] \geq DC_{thd} \end{cases}$$

$RLE[0] \rightarrow \text{coef. DC}$

$$u_{RLE}[1] = 0, \quad u_{RLE}[2] = 0, \quad N_{RLE} = 3$$

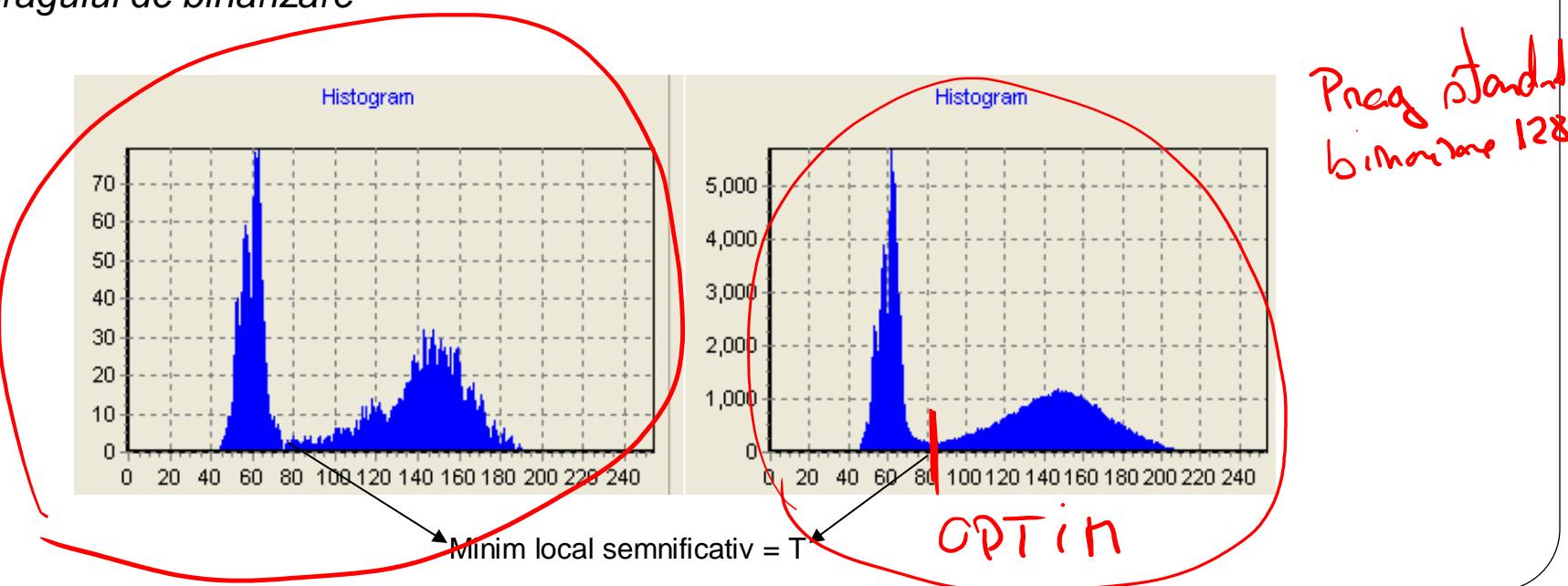
$EOB(0,0)$
 $\text{coeficienți AC} = 0$

- Dacă $E_{AC} > e_{thd}$, blocul are un conținut semnificativ de detaliu și trebuie decomprimat - binarizarea fiecarui pixel în parte:

$$b(i, j) = \begin{cases} 0, & \text{dacă } u(i, j) < T \\ 255, & \text{dacă } u(i, j) \geq T \end{cases} \quad i = 0, 1, \dots, 7, \quad j = 0, 1, \dots, 7$$

Selectia pragului de binarizare fara decompresie:

- Procedura curenta (la prelucrarea in domeniul spatial): examinarea histogramei; pragul de binarizare \Leftrightarrow un minim local in histograma
- În domeniul comprimat, nu avem acces la valorile luminantelor pixelilor pentru a realiza histograma imaginii
 \Rightarrow putem construi o histograma aproximativa utilizand coeficientii DC,
 \Leftrightarrow o aproximare a histogramei nivelerelor de gri, cu pierderea detalilor, dar *pastrarea minimelor si maximelor locale semnificative*
 \Rightarrow *histograma coeficientilor DC ai blocurilor ofera un bun indicator vizual pentru selectia pragului de binarizare*



Imagine binarizată în domeniul comprimat

blocuri cu
energie mare sunt dezcomprimate

Rezultate experimentale

Rezultatele pentru o valoare $e_{thd} = 0.1$

Imagine	EffBlock s [%]	Numărul de comparații cu prag pe imagine		MSE [%]
		Binarizare clasică	Binarizare cu algoritmul propus	
Img1.jpg	13.5	325.071	414.720	1.45
Img2.jpg	16.07	86.016	40.593	0.48
Img3.jpg	24	96.768	46.179	0.012
Img4.jpg	18.24	262.144	173.881	1.45

Imagine binarizată în domeniul spațial

Diferența dintre cele două rezultate



Imagine binarizată adaptiv
în domeniul comprimat



Imagine binarizată în
domeniul spațial

Algoritmii neliniari de prelucrare a imaginilor în domeniul comprimat

- Pt. majoritatea algoritmilor neliniari e dificilă transpunerea lor în domeniul coeficienților DCT, dar odată ce combinația corectă este găsită (pentru reformulare în domeniul comprimat) algoritmul este mult mai rapid.
- Chiar și dacă forma transpusă în domeniul comprimat este de multe ori mai complexă, este totuși mult mai rapidă:
 - Unu, se evită decompresia la nivel de pixel, procesarea pixelilor, recompresia.
 - Doi, se procesează mult mai puține date, deoarece după cuantizare majoritatea coeficienților (cei care aparțin frecvențelor înalte) devin zero

- Ex: Calculul pătratelor pixelilor dintr-o imagine:
(se folosește formula de convoluție în domeniul comprimat)

$$\begin{aligned} U_{dct,sq}(x_1, x_2) &= \frac{1}{4 \cdot Q(x_1, x_2)} \sum_i \sum_j C(i, x_1) \cdot C(j, x_2) \cdot [u(i, j)]^2 = \\ &= \sum_{y_1, y_2, w_1, w_2} U_{dct}(y_1, y_2) \cdot U_{dct}(w_1, w_2) \cdot W_Q(y_1, y_2, w_1, w_2, x_1, x_2) \end{aligned}$$

unde :

$$W_Q(y_1, y_2, w_1, w_2, x_1, x_2) = \frac{Q(y_1, y_2) \cdot Q(w_1, w_2)}{256 \cdot 64 \cdot Q(x_1, x_2)} W(x_1, y_1, w_1) \cdot W(x_2, y_2, w_2)$$

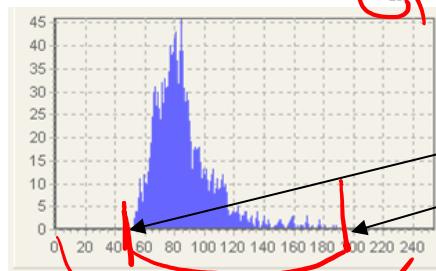
cu : $W(x, y, w) = \sum_i C(i, x) \cdot C(i, y) \cdot C(i, w)$

$$C(i, x) = A(x) \cos \frac{(2 \cdot i + 1) \cdot x \cdot \pi}{16}; \quad A(x) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } x = 0 \\ 1, & \text{for } x \neq 0 \end{cases}$$

Îmbunătățirea imaginilor folosind operatorul fuzzy de intensificare a unei funcții de apartenență.

- Pentru a reformula acest algoritm în domeniul comprimat avem nevoie de următoarele operații:
 - liniare: adunare, înmulțire cu o constantă, pătratul fiecărui pixel din imagine (calculat cu formula de conoluție în domeniul comprimat prezentată pe slide-ul anterior)
 - neliniare: comparare cu prag

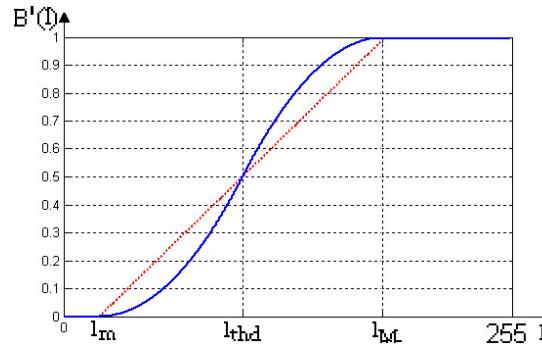
a. Funcția de apartenență pentru reprezentarea dinamicii nivelelor de gri din imagine prin conceptul "Luminos"



$$[l_{\min}, l_{\max}] \subseteq [l_m, l_M],$$

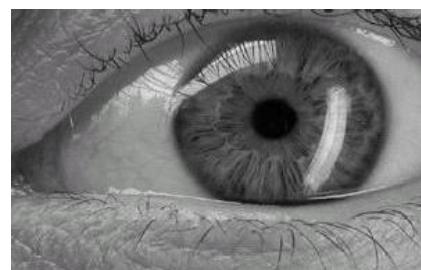
l - nivel de gri

$$B'(l) = INT(B(l)) = \begin{cases} 2 \cdot (\alpha \cdot l + b)^2, & \text{dacă } 0 \leq \alpha \cdot l + b \leq 0.5 \\ 1 - 2 \cdot (-\alpha \cdot l + 1 - b)^2, & \text{dacă } 0.5 \leq \alpha \cdot l + b \leq 1 \end{cases}$$



b. Rezultatul aplicării operatorului fuzzy de intensificare asupra funcției de apartenență "Luminos" – reprezentând noua dinamică a nivelelor de gri, în imaginea prelucrată

Rezultatul aplicării operatorului fuzzy de intensificare asupra $B(l)$ este funcția de apartență:



Algoritm de îmbunătățire a imaginilor bazat pe reguli fuzzy

Algoritmul de îmbunătățire a imaginilor bazat pe reguli fuzzy, Takagi-Sugeno, este un algoritm cu efect vizual foarte bun, dar ca orice algoritm fuzzy este neliniar, reformularea lui în domeniul comprimat nefiind directă.

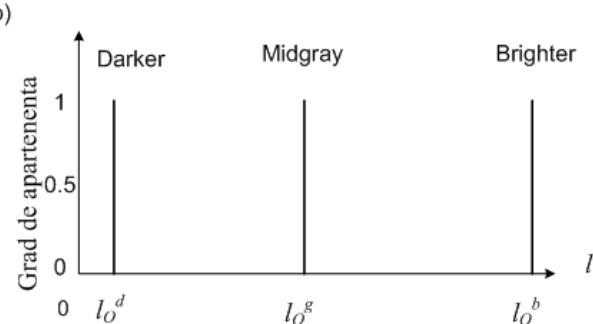
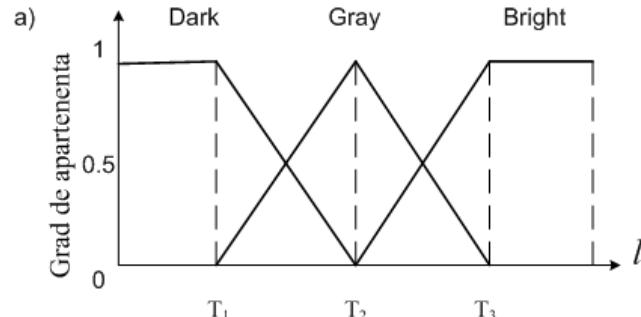
Sistemul de inferenta fuzzy Takagi-Sugeno cuprinde următoarele 3 reguli :

R1: DACĂ I_u este Dark ATUNCII I_v este Darker

R2: DACĂ I_u este Gray ATUNCII I_v este Midgray

R3: DACĂ I_u este Bright ATUNCII I_v este Brighter,

Funcțiile de apartenență de la intrarea și ieșirea sistemului Takagi-Sugeno sunt:



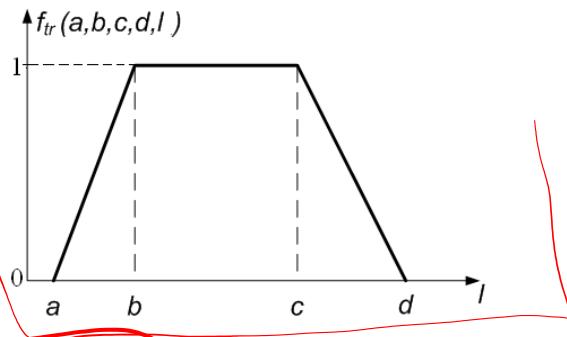
Pentru fiecare intensitate l_u^* de la intrarea sistemului fuzzy, în imaginea de ieșire, intensitatea corespunzătoare l_o^* se obține prin aplicarea mecanismului de inferență Takagi-Sugeno:

$$l_v^* = \frac{\mu_{Dark}(l_u^*) \cdot l_o^d + \mu_{Gray}(l_u^*) \cdot l_o^g + \mu_{Bright}(l_u^*) \cdot l_o^b}{\mu_{Dark}(l_u^*) + \mu_{Gray}(l_u^*) + \mu_{Bright}(l_u^*)}$$

unde: $\mu_{Dark}(l_u^*)$, $\mu_{Gray}(l_u^*)$, $\mu_{Bright}(l_u^*)$ sunt gradele de apartenență ale intensității curent procesate la multimile fuzzy definite peste intrare: *Dark, Gray și Bright*

Reformularea inferenței fuzzy în domeniul comprimat

Forma grafică și analitică a funcției pentru calculul gradelor de apartenență:



Algoritm clasic

$$f_{tr}(a, b, c, d, l) = \begin{cases} 0, & \text{dacă } l \in [0, a] \\ \frac{l-a}{b-a}, & \text{dacă } l \in [a, b] \\ 1, & \text{dacă } l \in [b, c] \\ \frac{-l+d}{d-c}, & \text{dacă } l \in (c, d] \\ 0, & \text{dacă } l \in (d, 255] \end{cases}.$$

Algoritm propus în domeniul comprimat

$$\begin{aligned} K_s &= [k_1^s, k_2^s] = f_{tr}^{DCT}(a, b, c, d, u_Q(0,0)) = \\ &= \begin{cases} [0 \ 0], & \text{dacă } u_Q(0,0) \in [-128, a-128) \\ \left[\frac{1}{b-a} \ \frac{128-a}{b-a} \right], & \text{dacă } u_Q(0,0) \in [a-128, b-128) \\ [1 \ 1], & \text{dacă } u_Q(0,0) \in [b-128, c-128) \\ \left[\frac{-1}{d-c} \ \frac{-128+d}{d-c} \right], & \text{dacă } u_Q(0,0) \in [c-128, d-128) \\ [0 \ 0], & \text{dacă } u_Q(0,0) \in [d-128, 128] \end{cases}. \end{aligned}$$

Blocul obținut prin aplicarea mecanismului de inferență Takagi-Sugeno în domeniul comprimat:

$$U_{Q,Int} = D_Q^{Dark} \cdot (l_o^d - 128) + D_Q^{Gray} \cdot (l_o^g - 128) + D_Q^{Bright} \cdot (l_o^b - 128).$$

unde: $D_q^s = c_m \cdot U_q + C_a^{dct}$, $s \in \{Dark, Gray, Bright\}$

$$C_a^Q = [k_2^s \ 0 \ 0 \ \dots \ 0], \quad c_m = k_1^s$$

Rezultate experimentale

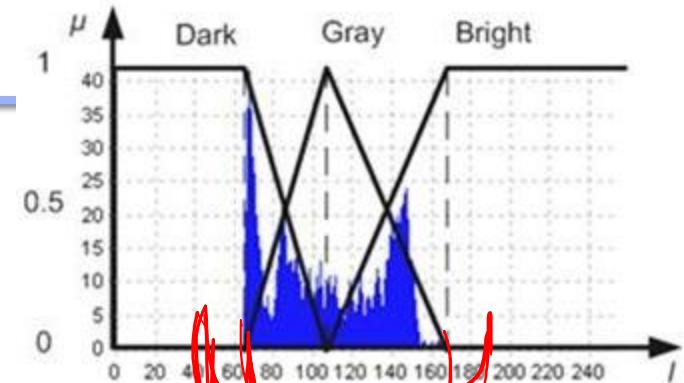
- Algoritmul este aplicat doar pe componenta de luminanță, dar poate fi folosit și pentru accentuarea imaginilor color, cu păstrarea componentelor de crominanță nemodificate



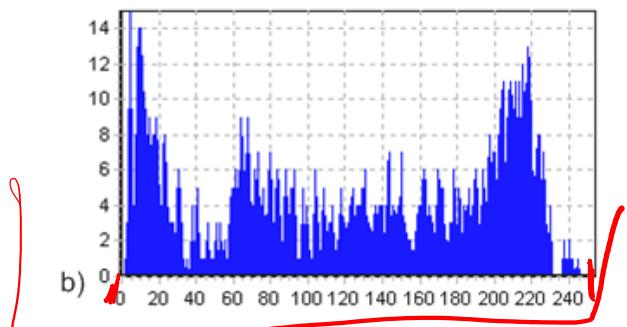
Imaginea originală



Imaginea după aplicarea algoritmului propus



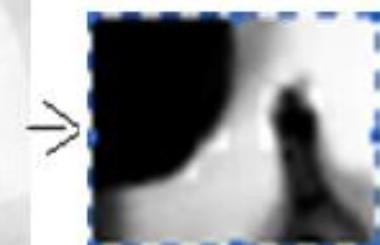
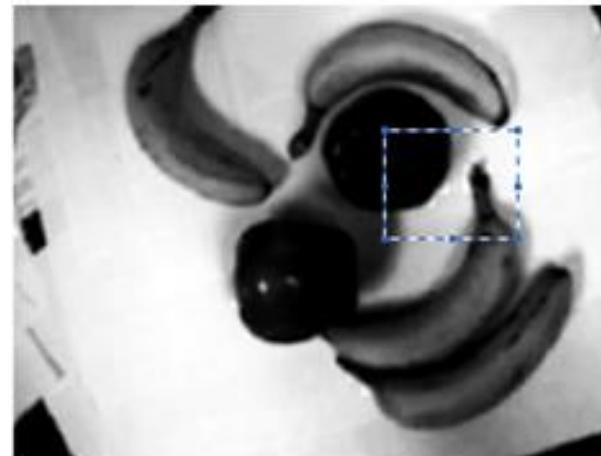
Funcțiile de apartenență de la intrarea sistemului fuzzy și histograma DC a componentei Y pentru imaginea originală



Histograma DC a componentei Y după accentuarea contrastului

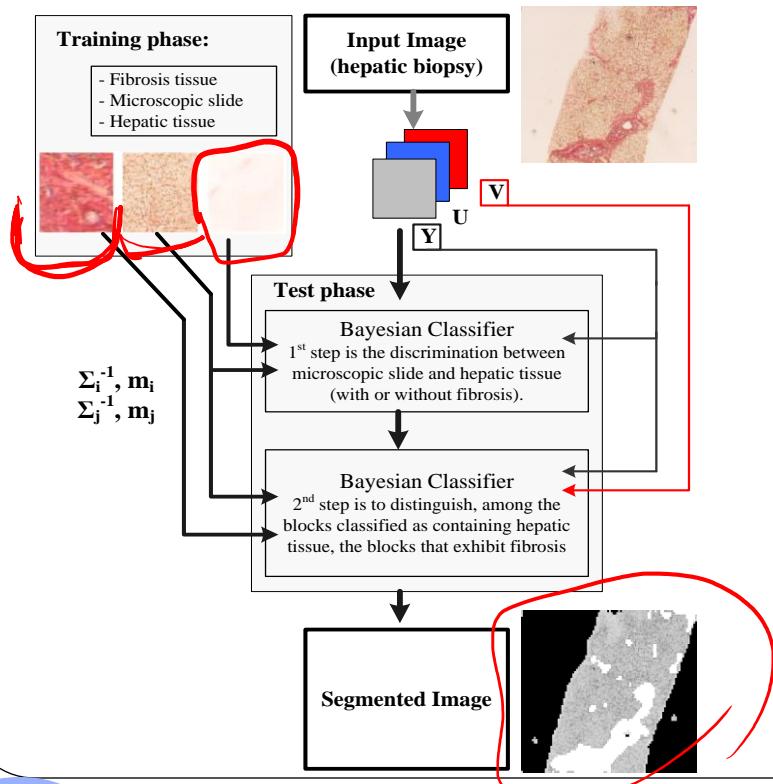
Probleme care apar la clasificarea tuturor pixelilor din bloc cu coeficientul DC

- Ambele algoritme fuzzy prezentate mai sus implică o (operatorul fuzzy INT) sau mai multe (alg. bazat pe reguli fuzzy) comparații cu prag
- Dacă se clasifică toți pixelii dintr-un bloc, pentru procesare, luând în considerare doar valoarea DC a acelui bloc, este foarte probabil că va apărea efectul de “blocking”
- Acest efect apare în cazul blocurilor care conțin muchii, există variații mari între valorile intensităților conținute
- Pentru evitarea acestui efect dar păstrarea avantajului de procesare a datelor direct în domeniul comprimat, se recomandă decompresia la nivel de pixel doar a blocurilor cu energie mare (variații mari) concentrată în coeficienții DCT, procesarea acestor blocuri la nivel de pixel și recompresia.
- Totuși, numărul acestor blocuri care vor fi decomprimează va fi sub 20%



Segmentarea imaginilor folosind clasificatorul Bayes și modelul Gaussian direct în domeniul comprimat

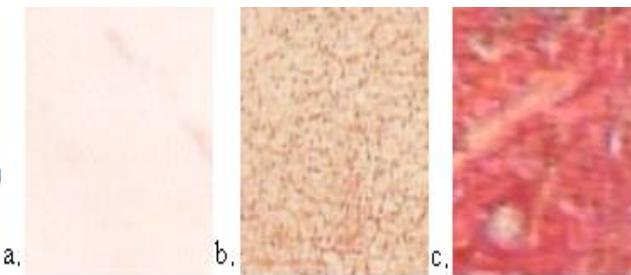
- Un tasc mai dificil este implementarea unor algoritmi care necesită învățare, sau implementarea unor clasificatoare mai complexe, direct în domeniul comprimat.
- Aceasta abordare își propune să fie o rapidă și eficace perspectiva pentru identificarea și cuantificarea fibrozei hepatice din biopsiile de ficat folosind metoda de clasificare a țesuturilor, clasificatorul Bayesian.



Aceasta abordare se bazează pe informația din culori la nivel de pixel, dar și din informația texturilor locale în blocurile vecine de 8x8 pixeli, făcând uz de informațiile existente deja în imaginile microscopice luând în considerare reprezentarea lor în formatul JPEG.

Formatul de compresie a imaginii în sine oferă informațiile necesare unei identificări și segmentări de o acuratețe ridicată a biopsiilor hepatice în țesut vs. lama și mai departe, țesut în țesut sănătos vs. fibros hepatică.

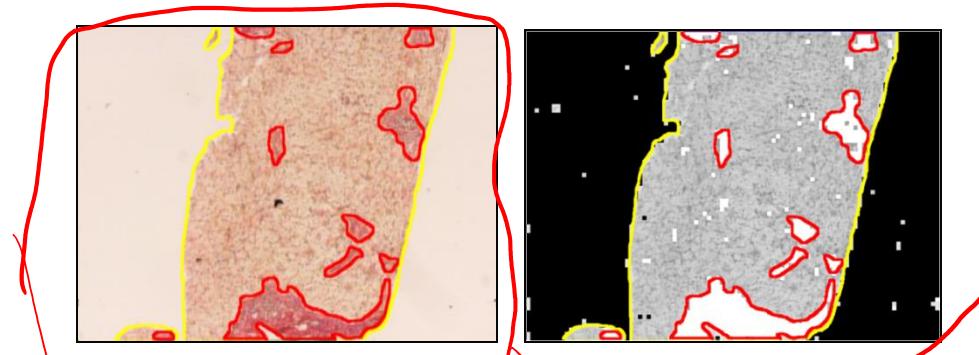
- Rez. Exp. ale metodei demonstrează nu doar o ridicată acuratețe, dar și o clasificare rapidă, datorată domeniului în care avem datele, lucru care permite o clasificare la nivel de bloc în detrimentul unei clasificări la nivel de pixel
 - ⇒ fazele de învățare și de clasificare sau test sunt semnificativ mai rapide.
 - ⇒ acuratețea oferita de folosirea informațiilor din textura este superioara uneia bazata doar pe informația din culorile la nivel de pixel.



Vedere microscopică a biopsiei ficatului: a. Lamă; b. Ţesut sănătos; c. Ţesut fibros



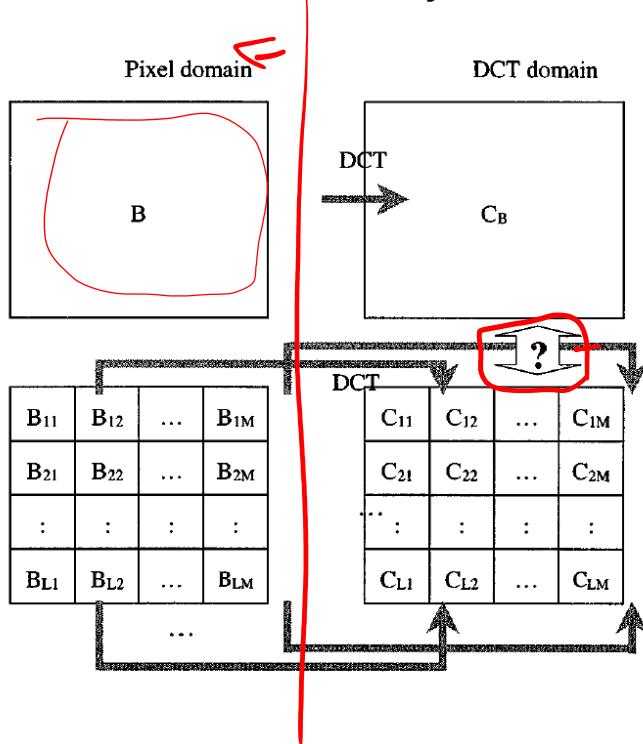
Imaginea hepatică: a) Originală; b) Segmentată la nivel de pixel în spațiul color RGB; c) Segmentată în domeniul comprimat la nivel de bloc DCT;



Diferențele dintre rezultatul segmentării cu algoritmul propus și „ground truth”

Relația spațială a coeficienților DCT dintre un bloc și sub-blocurile acestuia

- Fiecare element $C_{i,j}$ reprezintă setul de coeficienți DCT pentru sub-blocul $SB_{i,j}$ și implicit este o matrice cu $N \times N$ elemente.
- Tot mai mulți algoritmi de procesare de imagini sunt dezvoltăți în domeniul comprimat pentru a reduce calculele și pentru îmbunătățirea vitezei de procesare, o nouă problemă poate apărea din faptul că trebuie folosite blocuri DCT de dimensiuni diferite pentru a asigura performanțe optimizate:
 - extragerea trăsăturilor globale (blocuri mari), sau
 - extragerea trăsăturilor locale (blocuri mai mici)
- Derivarea directă a coeficienților DCT în subblocuri cu dimensiuni diferite este posibilă:



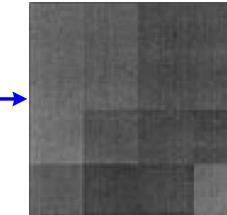
$$\boxed{C_B} = \frac{1}{M} \boxed{A^*} \begin{pmatrix} C_{0,0} & \cdots & C_{0,M-1} \\ \vdots & \ddots & \vdots \\ C_{L-1,0} & \cdots & C_{L-1,M-1} \end{pmatrix} \boxed{A^{*T}}$$

$$\begin{pmatrix} C_{0,0} & \cdots & C_{0,M-1} \\ \vdots & \ddots & \vdots \\ C_{L-1,0} & \cdots & C_{L-1,M-1} \end{pmatrix} = M A^{*-1} \boxed{C_B} A^{*T-1}$$

Exemplu de obținere a coeficienților DCT pe baza relațiilor între coeficienții DCT ai unui bloc de 4×4 pixeli și coeficienții DCT ai celor 4 sub-blocuri de 2×2 pixeli din bloc



Imaginea sursă



Blocul de 4×4 pixeli decupat

Transformarea din 4 sub-blocuri adiacente de 2×2 pixeli în blocul corespunzător de 4×4 pixeli:

106	97	83	85
106	95	84	85
105	84	74	69
77	60	57	89

Blocul cu luminanțele pixelilor

202	10	168	-1
1	-1	84	85
163	19	144	-13
26	2	-1	19

Coef. DCT a 4 blocuri de dimensiune 2×2

1	0	1	0
0.9239	0.3827	-0.9239	0.3827
0	1	0	-1
-0.3827	0.9239	0.3827	0.9239

Matricea A^*

339	23	22	-3
34	13	-13	0
-12	-16	8	-5
-1	13	-4	4

Transformarea dintr-un bloc de 4×4 pixeli în 4 sub-blocuri adiacente de 2×2 pixeli:

106	97	83	85
106	95	84	85
105	84	74	69
77	60	57	89

Blocul cu luminanțele pixelilor

339	23	22	-3
34	13	-13	0
-12	-16	8	-5
-1	13	-4	4

Coef. DCT a blocului de dimensiune 4×4

202	10	168	-1
1	-1	84	85
163	19	144	-13
26	2	-1	19

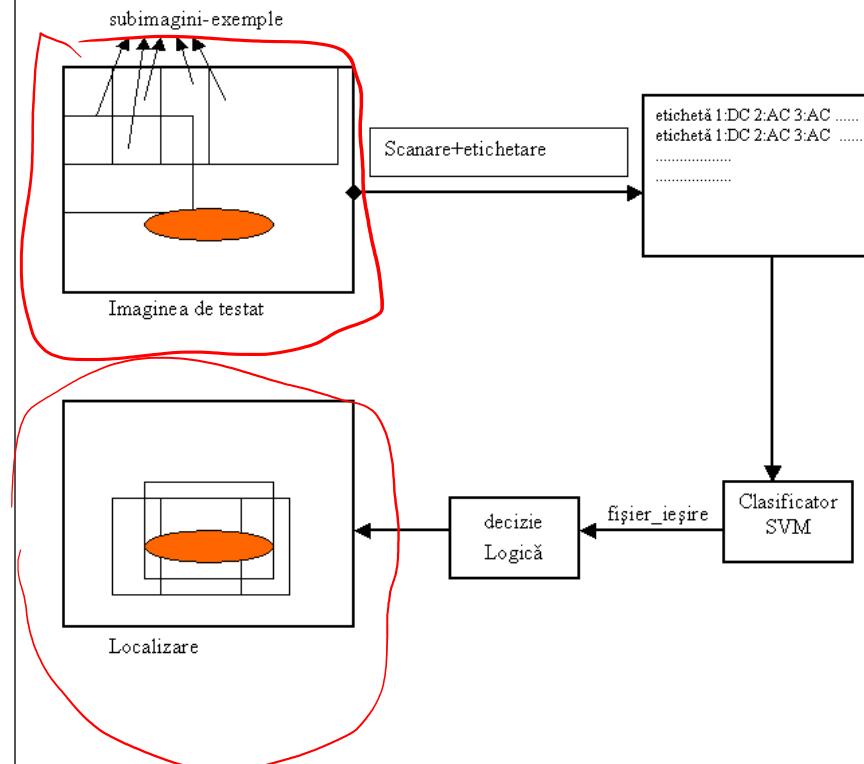
Matricea inversă A^*

Aplicarea mașinilor cu vectori suport în recunoașterea vizuală a obiectelor în domeniul comprimat

- Problema adresată
 - Implementarea unui clasificator instruibil direct în domeniul comprimat (mașini cu vectori suport)
 - probleme apărute:
 - în domeniul comprimat avem acces direct doar la blocuri de 8×8 pixeli – dimensiune prea restrictiva pentru regiunile de interes de analizat
- Abordarea
 - Utilizarea unui algoritm care oferă o relație directă pentru obținerea unui bloc de coeficienți DCT din sub-blocurile lui (și invers) – pentru setarea regiunii de interes direct în domeniul comprimat
- Avantaje
 - Se obțin regiunile de interes direct în domeniul comprimat fără necesitatea extragerii suplimentare a trasaturilor
 - O regiune de interes în domeniul comprimat va avea un număr redus de valori nenele – crește viteza de clasificare

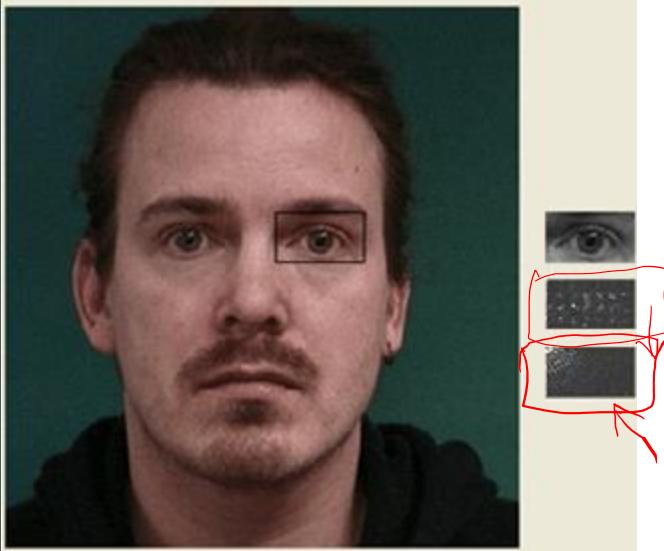
MVS direct în domeniul comprimat

- Mașinile cu vectori-suport sunt clasificatoare binare bazate pe învățarea statistică din exemple, care se bucură de un succes deosebit la ora actuală datorită performanțelor lor foarte bune în multe aplicații de analiză a imaginilor și recunoaștere a formelor.

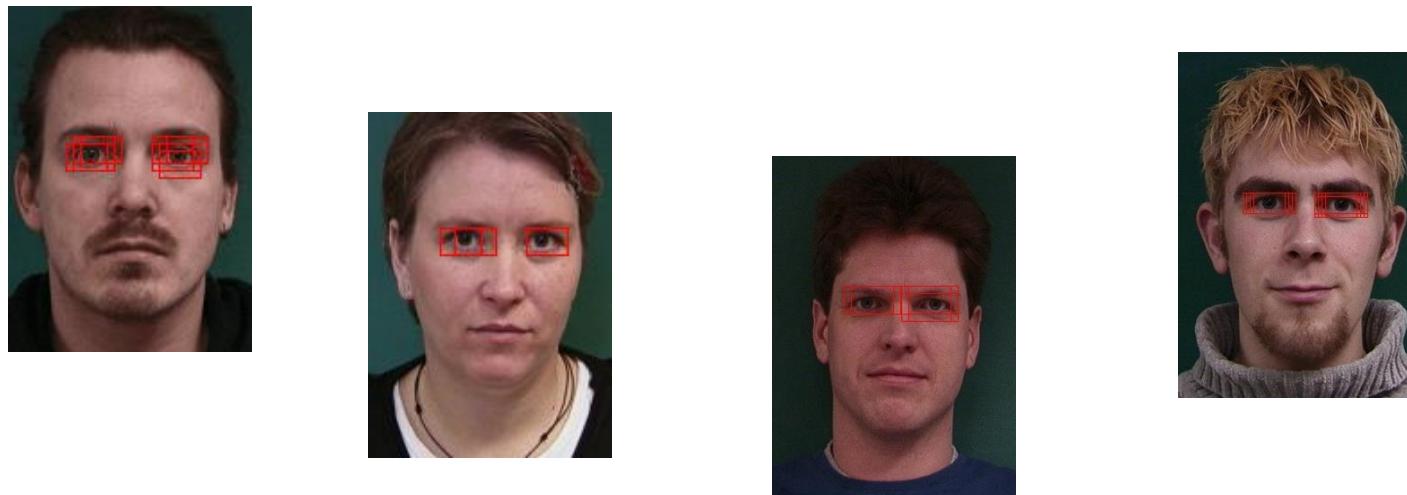


- Obiectul de recunoscut este o zonă de imagine (fereastră de dimensiuni cunoscute din faza de antrenare), descris prin coeficienții DCT (în cazul nostru) din fereastră scanați în ordine rând cu rând.
- În cazul procesării imaginilor JPEG direct în domeniul comprimat avem la dispoziție doar blocuri de 8×8 - de multe ori un ochi ocupă o zonă din imagine mai mare decât atât.
- După o analiză a relațiilor dintre un bloc de coeficienți DCT și sub-blocurile sale, s-a putut implementa o aplicație care ne oferă o relație directă pentru formarea din mai multe blocuri de coeficienții DCT un singur bloc conținând coeficienții DCT, fără decompresia imaginilor și aplicarea transformatei DCT pe întregul bloc.

Aplicarea mașinilor cu vectori suport în recunoașterea vizuală a obiectelor în domeniul comprimat



- pentru a reduce costul de procesare și de memorie
 - operațiile implicate în analiza imaginilor: extragerea trăsăturilor, selecția regiunii de interes, clasificarea și localizarea obiectelor de interes – realizate direct în domeniul comprimat
- Aplicație: localizare exactă a regiunii ochilor - sistemele de urmărire a traiectoriei privirii
- Rezultatele sunt usor superioare clasificării în domeniul spatial



Problema JPEG

Fie $U_{dct}[8 \times 8]$ transformata DCT a blocului de 8×8 pixeli curent de codat folosind standardul JPEG și $Q[8 \times 8]$ matricea de cuantizare folosită, iar tabela de coduri pt. coeficientul de curent continuu cea din Fig. 1.

- a) Cum arată matricea coeficienților DCT cuantizați (valorile rezultate în urma cuantizării se vor rotunji la cel mai apropiat întreg)?
- b) Reprezentați șirul coeficienților DCT cuantizați obținut în urma unei ordonări în zig-zag. Care este rolul ordonării în zig-zag?
- c) Să se codeze RLC șirul coeficienților DCT cuantizați și ordonați în zig-zag.
- d) Dacă valoarea cuantizată a coeficientului de c.c. (coeficientul DC) din blocul anterior codat este -16, care este valoarea care va fi codată entropic pentru coeficientul de c.c. din blocul curent?
- e) Care este codarea utilizată în standardul JPEG pentru codarea entropică a datelor? Care este codul coeficientului de c.c. din blocul curent codat entropic?
- f) Care este forma analitică a operației de negativare a imaginilor direct în domeniul comprimat? Aplicați operația de negativare pe matricea coeficienților DCT cuantizați obținută la punctul (a).

-227,00	40,71	66,32	-0,21	5,75	-0,60	2,02	-0,94
-164,88	64,14	70,93	6,57	5,45	-1,96	-1,59	1,07
-37,63	20,70	19,44	4,69	3,06	3,36	0,91	2,22
4,28	-2,34	-1,03	-0,38	-3,50	-3,65	-6,25	2,13
2,25	-0,11	-5,07	-5,63	-4,50	-2,16	-1,15	-0,11
8,49	-7,23	-0,67	-7,00	-1,88	-0,44	4,34	-0,62
9,09	2,35	1,91	-4,28	-0,46	-0,79	1,06	1,51
6,02	3,72	2,34	-1,27	-3,17	2,18	1,18	-1,32

$U_{dct} =$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

$Q =$

Categorie	Codul de bază	Lungimea	Categorie	Codul de bază	Lungimea
0	010	3	6	1110	10
1	011	4	7	11110	12
2	100	5	8	111110	14
3	00	5	9	1111110	16
4	101	7	A	11111110	18
5	110	8	B	111111110	20

round

$$U_{dct_Q}(i,j) = \lfloor U_{dct}(i,j) / Q(i,j) \rfloor, \text{ unde } i, j = \{0 \dots 7\}$$

U_{dct} =

-227,00	40,71	66,32	-0,21	5,75	-0,60	2,02	-0,94
-164,88	64,14	70,93	6,57	5,45	-1,96	-1,59	1,07
-37,63	20,70	19,44	4,69	3,06	3,36	0,91	2,22
4,28	-2,34	-1,03	-0,38	-3,50	-3,65	-6,25	2,13
2,25	-0,11	-5,07	-5,63	-4,50	-2,16	-1,15	-0,11
8,49	-7,23	-0,67	-7,00	-1,88	-0,44	4,34	-0,62
9,09	2,35	1,91	-4,28	-0,46	-0,79	1,06	1,51
6,02	3,72	2,34	-1,27	-3,17	2,18	1,18	-1,32

$Q =$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

a)

$$U_{dct_Q}(0,0) = \left\lfloor \frac{-227}{16} \right\rfloor = -14$$

a)

-14	4	2	0	0	0	0	0
-14	5	1	0	0	0	0	0
-3	2	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Concentrate values
in storage bins

b) $2D \rightarrow 1D$
Scans in zig-zag

[14, 4, -14, -3, 5, 7, 0,
5, 2, 0, 0, 0, 1, 0, 0, ...]
EOB

{ Coeff DC \rightarrow DPCM }
{ Coeff AC \rightarrow RLE }

f) $U_{dct_Q,N} = (-1) \cdot U_{dct,Q}$
 \rightarrow facilitates calculations

c) Codice RLE 

$-14, \underline{(0, 4)}, (0, -14), (0, -3), (0, 5), (0, 7), (1, 5), (0, 2), (3, 1) \text{ (0)}$

\hookrightarrow codice Huffman

$\hookrightarrow DC \rightarrow$ codice dif. -DPCM

$$d) DC_{i-1} = -16 \quad ; \quad DC_i = -14$$

$$e = DC_i - DC_{i-1} = -14 - (-16) = 2 \quad \hookrightarrow \text{codice Huffman}$$

e) Codice Huffman cof. DC [cod. loss - ampl. val]

$$2_+ = \boxed{\log 10}$$

6. Transformata DCT a blocului curent de codat în standardul JPEG este cea din Fig. 6.a. Matricea de cuantizare folosită este cea din Fig. 6.b, iar tabela de coduri pt. coeficientul de current continuu – în Fig. 6.c.

- Cum arată șirul coeficienților DCT cuantizați și ordonați în zig-zag? Care este rolul ordonării în zig-zag?
- Dacă valoarea cuantizată a coeficientului de c.c. din blocul anterior codat este 16, care este codul coeficientului de c.c. din blocul curent?

200	69	-50	-24	0	16	21	14
147	-63	-45	-22	0	15	19	12
0	0	0	0	0	0	0	0
-52	22	16	8	0	-5	-7	-4
0	0	0	0	0	0	0	0
34	-15	-11	5	0	3	4	3
0	4	9	0	0	0	0	0
-29	12	-3	0	0	-2	0	0

Fig. 6.a

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Fig. 6.b

Categorie	Codul de bază	Lungimea	Categorie	Codul de bază	Lungimea
0	010	3	6	1110	10
1	011	4	7	11110	12
2	100	5	8	111110	14
3	00	5	9	1111110	16
4	101	7	A	11111110	18
5	110	8	B	111111110	20

Fig. 6.c