

Concepte și aplicații în Vederea Artificială

Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

Radu Ionescu

radu.ionescu@fmi.unibuc.ro

Curs optional
anul III, semestrul I, 2023-2024

Cursul trecut

1. Aspecte organizatorice legate de cursul de VA
 1. Ce este VA?
 2. Aplicații de succes în VA
 3. Formarea imaginilor
 4. Structura cursului de VA
 5. Bibiliografie
 6. Kahoot!

Examen – evaluare în iarnă

- În funcție de trei aspecte:
 - teme de laborator (vor fi două de-a lungul semestrului): T1, T2;
 - primiți T1 în săptămâna 7 (15 noiembrie, termen limită 3 decembrie), T1 va fi punctată cu 5 puncte (+ posibil bonus);
 - primiți T2 în săptămâna 12 (20 decembrie, termen limită 21 ianuarie), T2 va fi punctată cu 5 puncte (+ posibil bonus);
 - lucrare finală de laborator: L1;
 - lucrarea se dă în ultima săptămână din sesiune, 5-11 februarie 2024 și valorează exact 5 puncte;
 - bonusul de la curs (BC) și laborator (BL) – nu mai mult de 1 punct cumulat.

Examen – evaluare în iarnă

- puteți obține nota numai din cele două teme și bonusurile de la curs și laborator (verificare = 21 ianuarie):
 - nota $N1 = T1 + T2 + \min(1, BC + BL)$

SAU

- puteți obține nota numai din lucrarea finală de laborator (test pe calculator în ultima săptămână 5-11 februarie din sesiunea de iarnă) + bonusul de la curs:
 - nota $N2 = 2*L1 + \min(1, BC)$

SAU

- puteți obține nota din lucrarea finală de laborator (test pe calculator în ultima săptămână 5-11 februarie din sesiunea de iarnă) și o notă din temă + bonusul de la curs:
 - nota $N3 = L1 + \max(T1, T2) + \min(1, BC)$
- Nota finală = $\min(10, \max(N1, N2, N3))$

Examen – evaluare în iarnă

- Nota finală = $\min(10, \max(T1 + T2 + \min(1, BC + BL), 2*L1 + \min(1, BC), L1 + \max(T1, T2) + \min(1, BC)))$
- nu există praguri minimale de îndeplinit

Restanță + reexaminare + mărire - evaluare

- puteți obține nota numai din lucrarea finală de laborator (test pe calculator în sesiune) + bonusul de la curs.
 - nota $N4 = 2*L2 + \min(1, BC)$

SAU

- puteți obține nota din lucrarea finală de laborator (test pe calculator în sesiune) și o notă din temă (trebuie să aveți cel puțin o temă rezolvată) + bonusul de la curs.
 - nota $N5 = L2 + \max(T1, T2) + \min(1, BC)$
- Nota finală = $\min(10, \max(N4, N5))$

Restanță pentru 2024/2025

- nu ținem cont de niciun punctaj din anul curent 2023/2024.

Rotunjirea notelor obținute

- orice notă $x < 5$ se rotunjește la partea întreagă $[x]$
 - nota 4.99 se rotunjește la nota 4;
 - nota 3.4 se rotunjește la nota 3.
- pentru orice notă $x \geq 5$:
 - dacă partea fractionară $\{x\} \geq 0.5$ atunci rotunjim la $[x] + 1$
 - nota 9.5 se rotunjește la nota 10;
 - dacă partea fractionară $\{x\} < 0.5$ atunci rotunjim la $[x]$
 - nota 9.45 se rotunjește la nota 9;

Bonusul de la curs + laborator

- puteți acumula maxim 1 punct bonus de la curs + laborator;
- bonus de la laborator = 0,05p o prezență la laborator (maxim o prezență pe săptămână);
- bonusul de la curs:
 - la sfârșitul fiecărui curs veți primi întrebări din materia predată la acel curs;
 - vom folosi platforma Kahoot!
 - număr de întrebări flexibil în funcție de fiecare curs;
 - premiem la fiecare curs primii 9 studenți: primii 3 studenți iau 0.15p, următorii 3 studenți iau 0.10p, următorii 3 studenți iau 0.05p;

Tipuri de imagini digitale

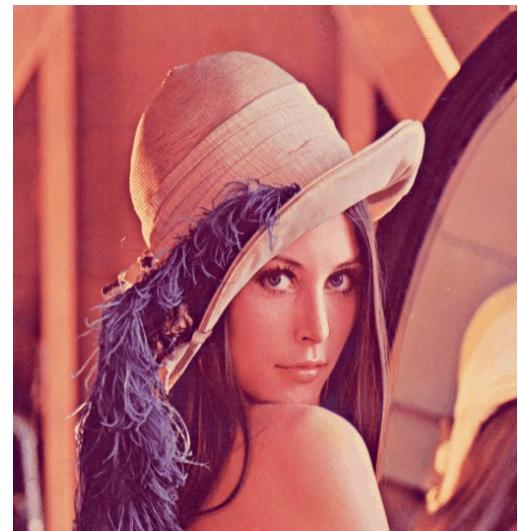
Binare



Grayscale
(tonuri de gri)



Color



Luminozitate	negru, alb	tonuri de gri	R G B
Valori	{0,1}	{0, ..., 255}	{0, ..., 255} ³
Culori	negru - 0, alb - 1	negru - 0, gri - 128, alb - 255	(255,0,0), (0,255,0), (0,0,255), (0,0,0), (255,255,255), (255,255,0), (255,125,0), (0,255,255), (255,0,255)
Memorie/pixe	1 bit/pixel	8 biți/pixel	24 biți/pixel

Tema 1 de anul trecut

Calculator automat de scor
pentru Scrabble

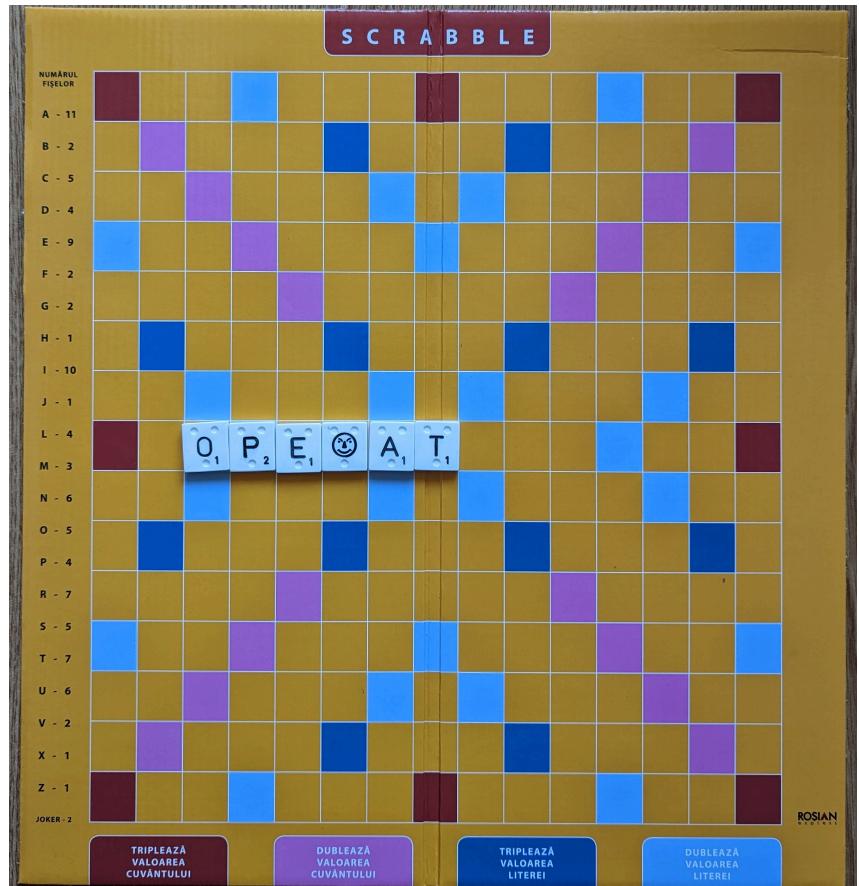
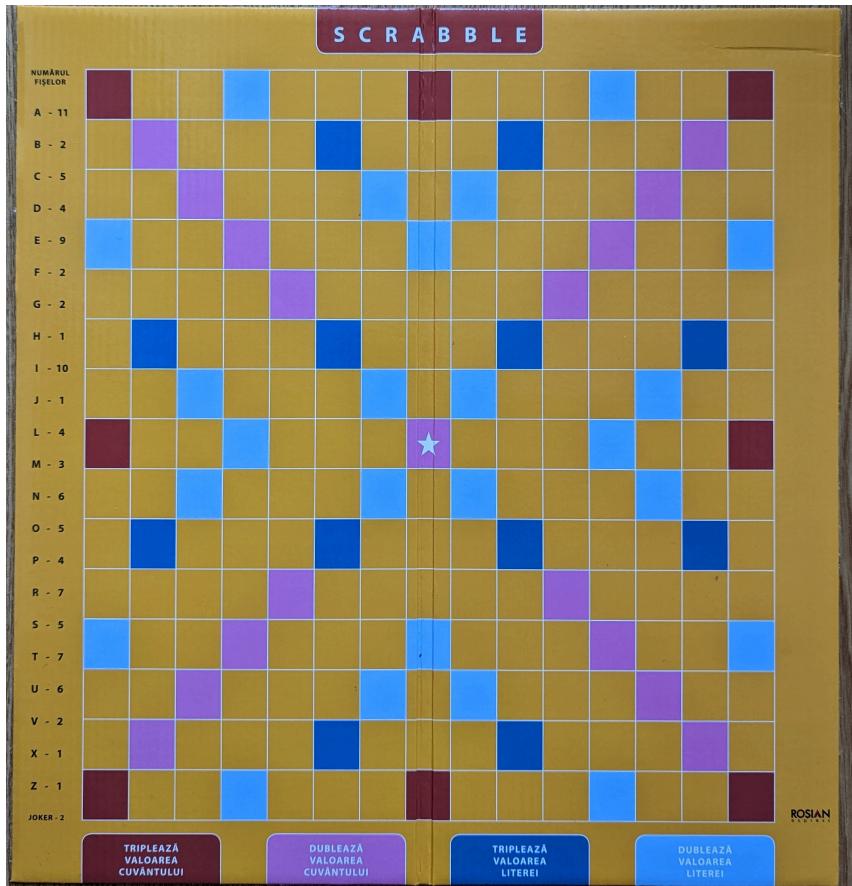
<https://tinyurl.com/CAVA-2022-TEMA1>

Scrabble

Scrabble este un joc în care participanții formează cuvinte prin plasarea de litere pe o tablă având dimensiunile de 15×15 pătrate (Figura 1). Cuvintele se pot forma pe orizontală sau pe verticală, precum la cuvinte încrucișate, iar punctajul obținut este mai mare atunci când literele folosite sunt mai rare (mai valoroase) sau când sunt plasate pe pătrate divers colorate care dau diferite bonificații ale tablei de joc. Cuvintele sunt valide doar dacă corespund dicționarelor acceptate oficial. Fiecare jucător concurează pentru a obține un scor cât mai mare folosind diferite combinații de litere plasându-le cât mai bine pe tablă astfel încât să profite de pătratele premium.



Exemplu de formare a cuvintelor + calcul scor



$$\text{OPE?AT: } (1(\text{O}) + 2(\text{P}) * 2(\text{SDL}) + 1(\text{E}) + 0(\text{R}) + 1(\text{A}) + 1(\text{T})) * 2(\text{SDC}) = 16$$

TOTAL: 16

SDL = SCOR DUBLU LITERA
STL = SCOR TRIPLU LITERA

SDC = SCOR DUBLU CUVANT
STC = SCOR TRIPLU CUVANT

Exemplu de formare a cuvintelor + calcul scor



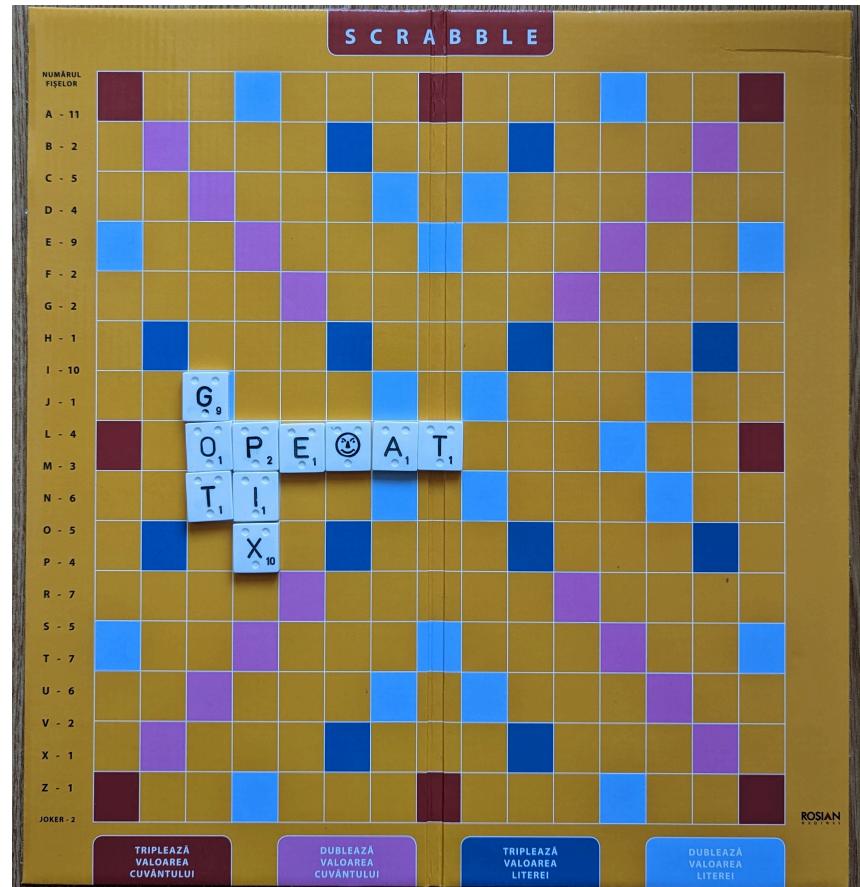
$$\text{PIX: } 2(\text{P}) + 1(\text{I}) + 10(\text{X}) = 13$$

TOTAL: 13

SDL = SCOR DUBLU LITERA
STL = SCOR TRIPLU LITERA

SDC = SCOR DUBLU CUVANT
STC = SCOR TRIPLU CUVANT

Exemplu de formare a cuvintelor + calcul scor



$$\text{GOT: } 9(\text{G}) * 2(\text{SDL}) + 1(\text{O}) + 1(\text{T}) * 2(\text{SDL}) = 21$$

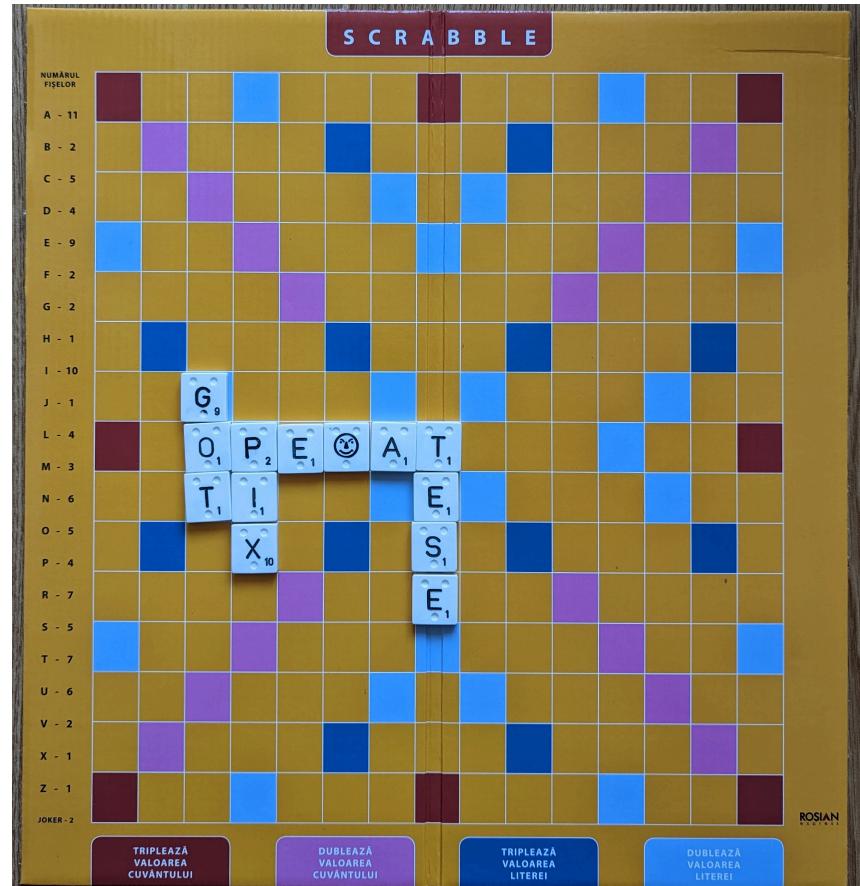
$$\text{TI: } 1(\text{T}) * 2(\text{SDL}) + 1(\text{I}) = 3$$

$$\text{TOTAL: } 21 + 3 = 24$$

SDL = SCOR DUBLU LITERA
 STL = SCOR TRIPLU LITERA

SDC = SCOR DUBLU CUVANT
 STC = SCOR TRIPLU CUVANT

Exemplu de formare a cuvintelor + calcul scor



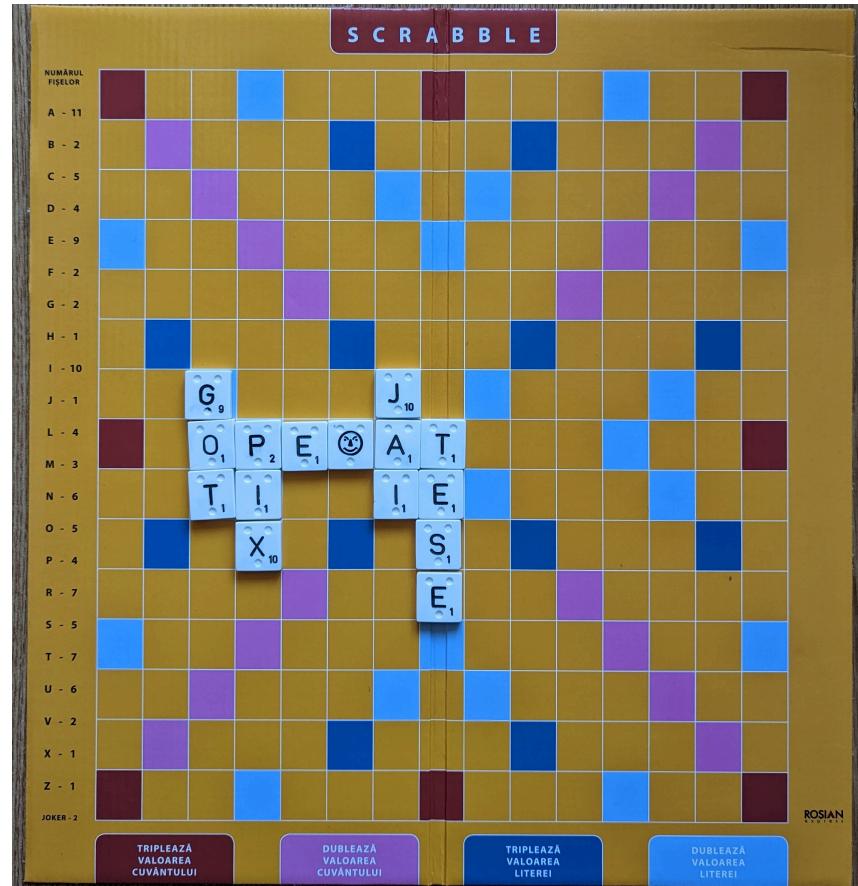
$$\text{TESE: } 1(T) + 1(E) + 1(S) + 1(E) = 4$$

TOTAL: 4

SDL = SCOR DUBLU LITERA
STL = SCOR TRIPLU LITERA

SDC = SCOR DUBLU CUVANT
STC = SCOR TRIPLU CUVANT

Exemplu de formare a cuvintelor + calcul scor



$$\text{JAI: } 10(\text{J}) * 2(\text{SDL}) + 1(\text{A}) + 1(\text{I}) * 2(\text{SDL}) = 23$$

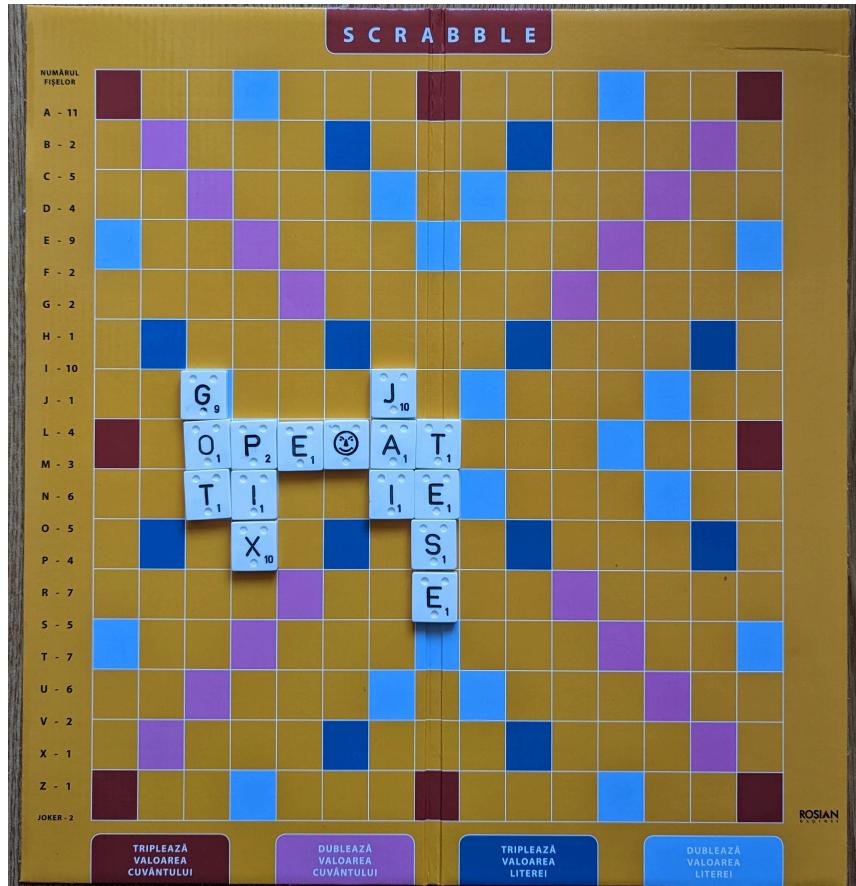
$$\text{IE: } 1(\text{I}) * 2(\text{SDL}) + 1(\text{E}) = 3$$

$$\textbf{TOTAL: } 23 + 3 = 26$$

SDL = SCOR DUBLU LITERA
STL = SCOR TRIPLU LITERA

SDC = SCOR DUBLU CUVANT
STC = SCOR TRIPLU CUVANT

Exemplu de formare a cuvintelor + calcul scor



COOPERAT: $(1(C) + 1(O) + 1(O) + 2(P) + 1(E) + 0(?=R) + 1(A) + 1(T)) * 3(STC) = 24$

TOTAL: 24

SDL = SCOR DUBLU LITERA
STL = SCOR TRIPLU LITERA

SDC = SCOR DUBLU CUVANT
STC = SCOR TRIPLU CUVANT

Tema 2 de anul trecut

Detectarea și recunoașterea facială a personajelor din serialul de desene animate

Viața cu Louie

<https://tinyurl.com/CAVA-2022-TEMA2>

Proiectul 2 de anul trecut - Viața cu Louie

Viața cu Louie este un serial de desene animate lansat în anul 1994 și care are 3 sezoane, a căte 13 episoade. Serialul prezintă întâmplări din viața lui Louie Anderson, un copil de 8 ani care trăiește în Cedar Knoll, Wisconsin (SUA). Multe episoade sunt inspirate din viața adevăratului Louie Anderson, comedian american care apare la începutul episoadelor și care le vorbește telespectatorilor, acesta regizând, de altfel, și toate episoadele.



Task 1 – detectare facială

Prima problemă pe care o aveți de rezolvat constă în detectarea facială a *tuturor* fețelor personajelor care apar în imagini. Pentru fiecare imagine de intrare algoritmul vostru trebuie să returneze o mulțime de detecții asociate (fereastră dreptunghiulară și scor) ce localizează *toate* fețele dintr-o imagine. Figura 1 arată câteva exemple din mulțimea de antrenare și adnotările corespunzătoare, ce constau în ferestre dreptunghiulare de culoare roșie ce încadrează perfect *fiecare față*.

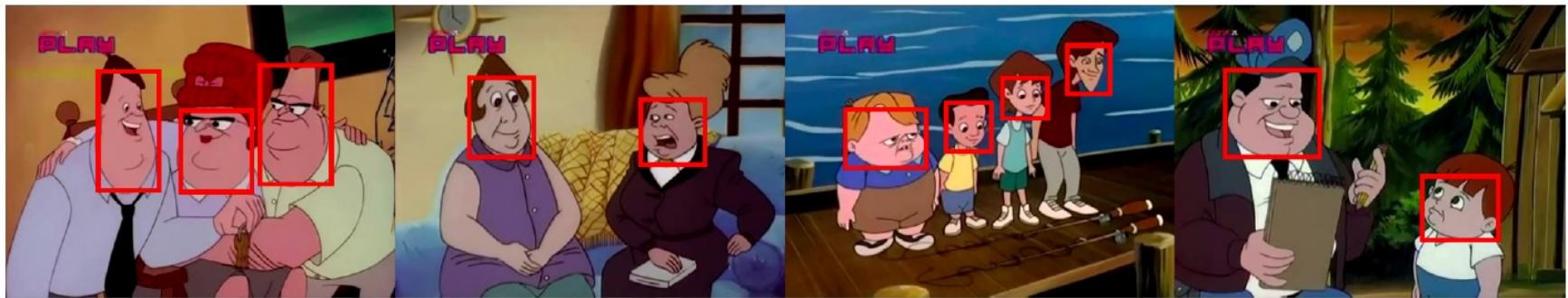


Figura 1: Detectare facială a personajelor din Viața cu Louie: fiecare față de interes este adnotată cu o fereastră dreptunghiulară de culoare roșie.

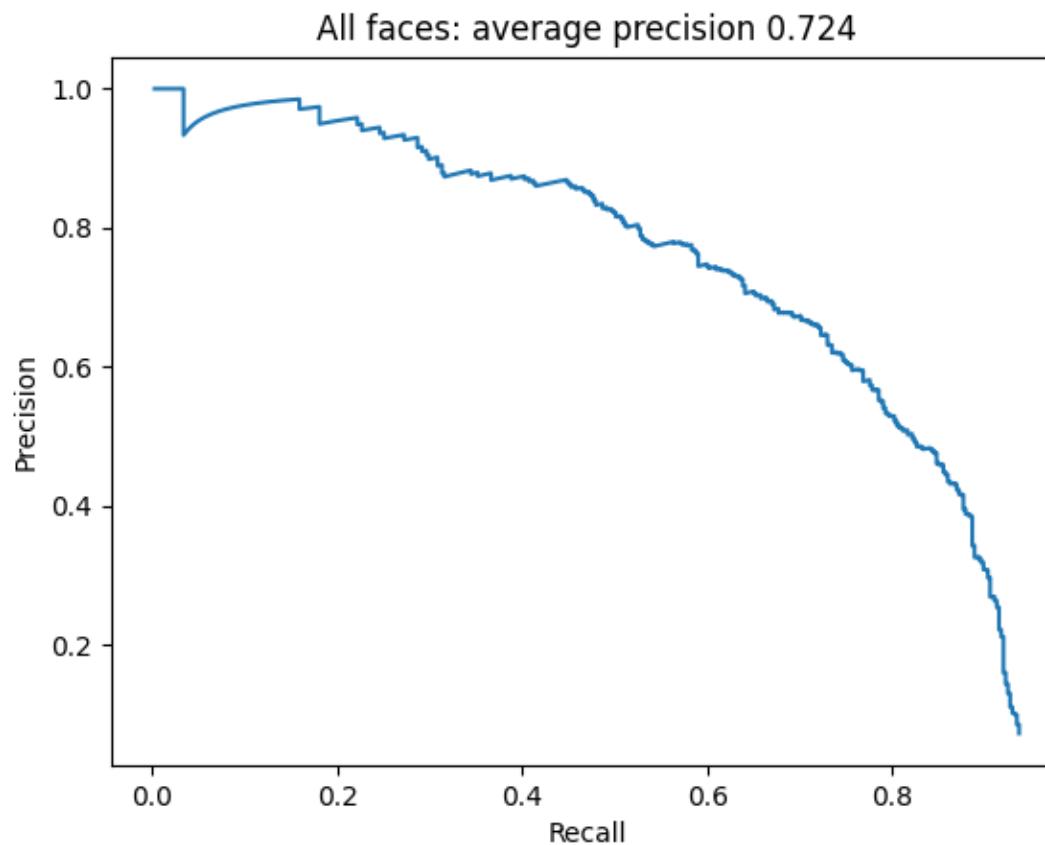
Task 2 – recunoaștere facială

A doua problemă pe care o aveți de rezolvat constă în recunoașterea facială a numai anumitor personaje. Alături de Louie, personajul principal din serial, apar cu preponderență alte trei personaje: Andy - tatăl său, Ora - mama sa, Tommy - fratele său. Vom considera problema recunoașterii faciale numai pentru aceste patru personaje, Andy, Ora, Louie și Tommy. Pentru fiecare imagine de intrare algoritmul vostru trebuie să returneze o mulțime de detecții asociate (numele personajului, fereastră dreptunghiulară și scor) ce localizează fețele celor patru personaje de interes (Andy, Ora, Louie, Tommy) din imagine. Figura 2 arată câteva exemple din mulțimea de antrenare și adnotările corespunzătoare, ce constau în ferestre dreptunghiulare ce încadrează perfect *fețele de interes* (pentru Andy, Ora, Louie și Tommy). Fiecare detecție are o culoare specifică clasei personajului (albastru - Andy, galben - Ora, verde - Louie, violet - Tommy).



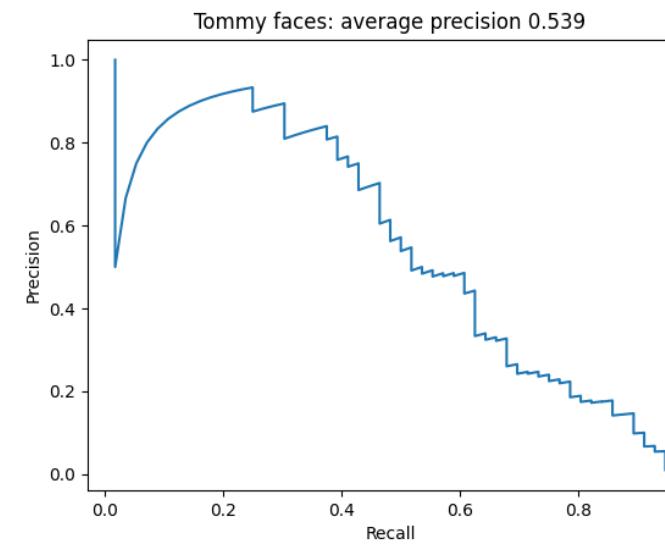
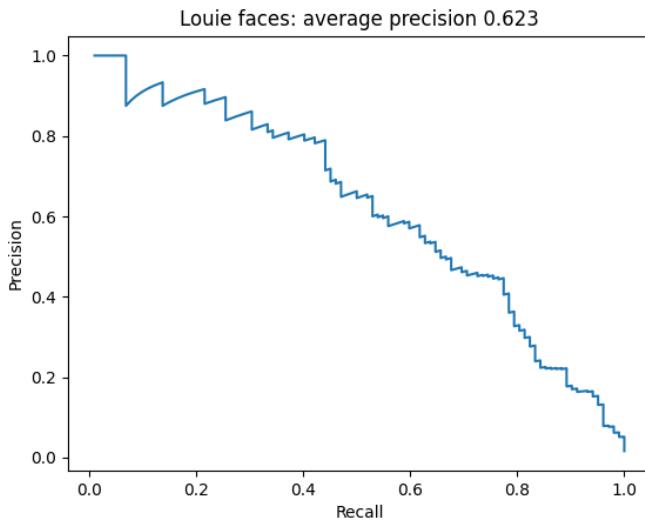
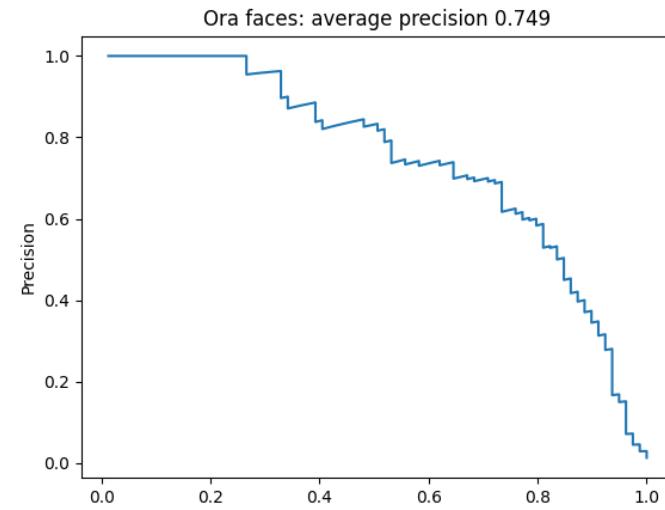
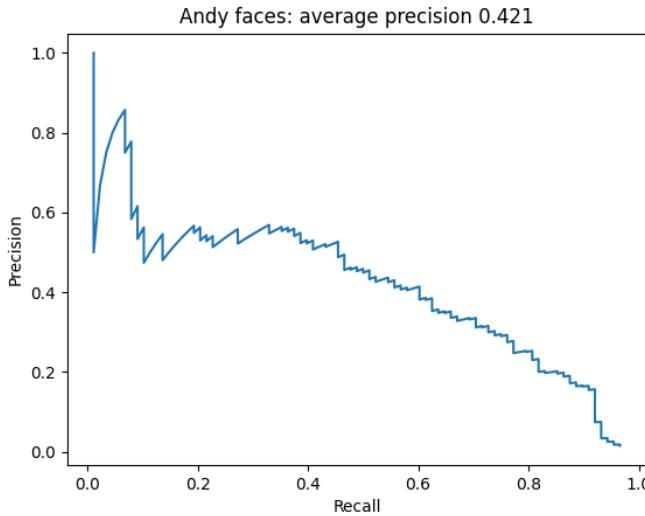
Figura 2: Recunoașterea facială a personajelor din *Viața cu Louie*: fiecare față de interes este adnotată cu o fereastră dreptunghiulară de culoare specifică clasei personajului (albastru - Andy, galben - Ora, verde - Louie, violet - Tommy).

Evaluare task 1 – AP pe datele de test



(performanța soluției noastre pe datele de validare = 200 de imagini)

Evaluare task 2 – mAP pe datele de test



(performanța soluției noastre pe datele de validare = 200 de imagini)

Cursul de azi

- Diverse modele pentru zgomot în imagini
 - salt and pepper, impuls
 - Gaussian (normal)
- Filtrarea liniară
 - corelație, conoluție
 - filtre: de medie, Gaussian, accentuare
 - aplicație: imagini hibrid
- Filtrarea neliniară
 - filtrul median



normal

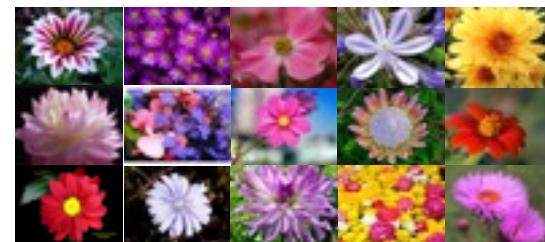


Laborator 2+3: Imagini mozaic

imagine de referință



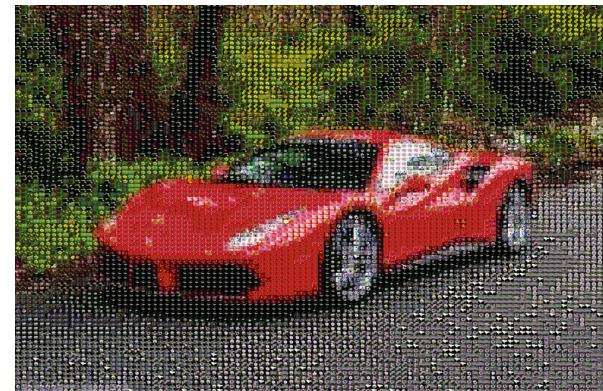
colectie de imagini (piese)
de dimensiuni reduse
 28×40 pixeli

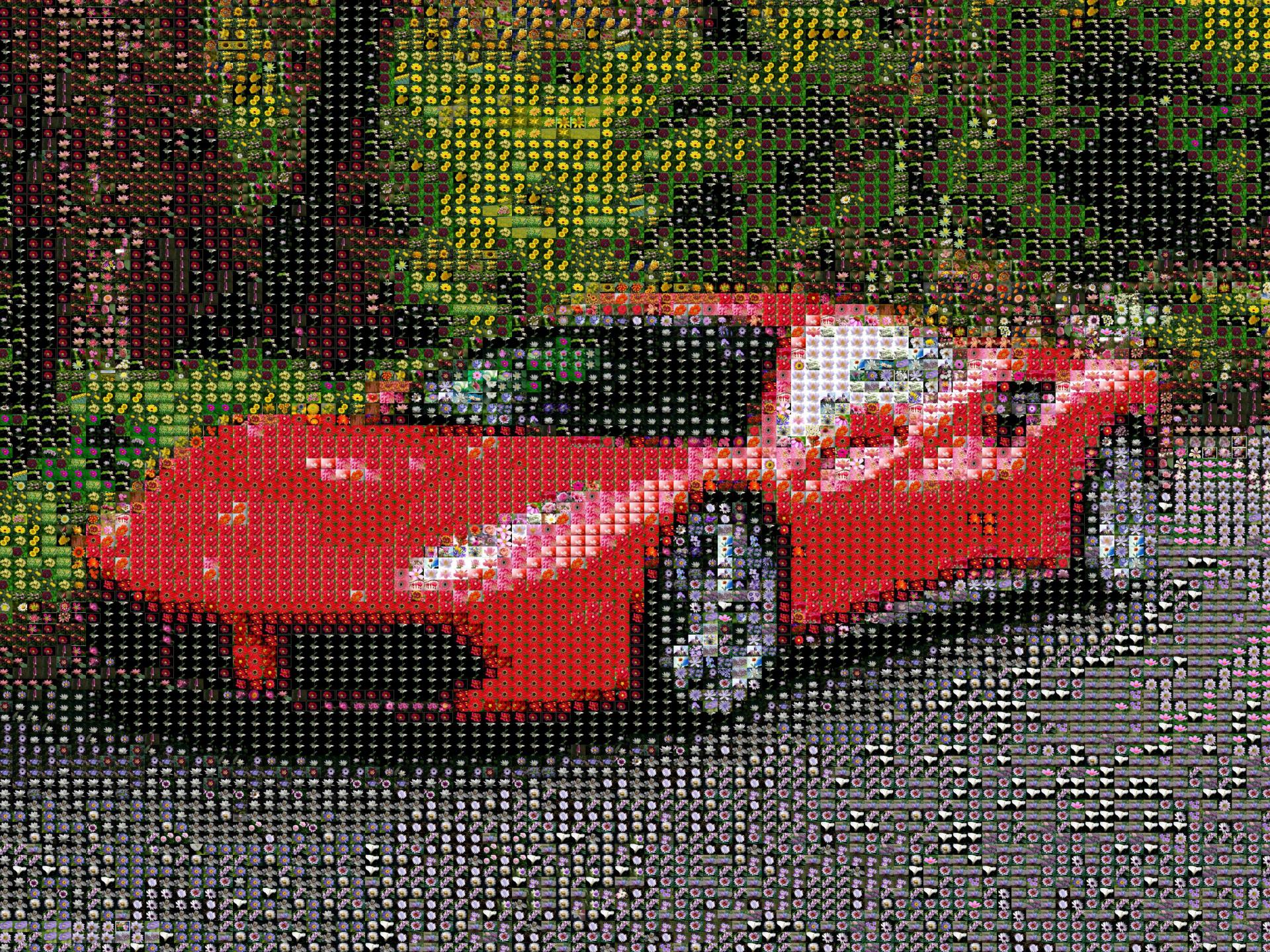


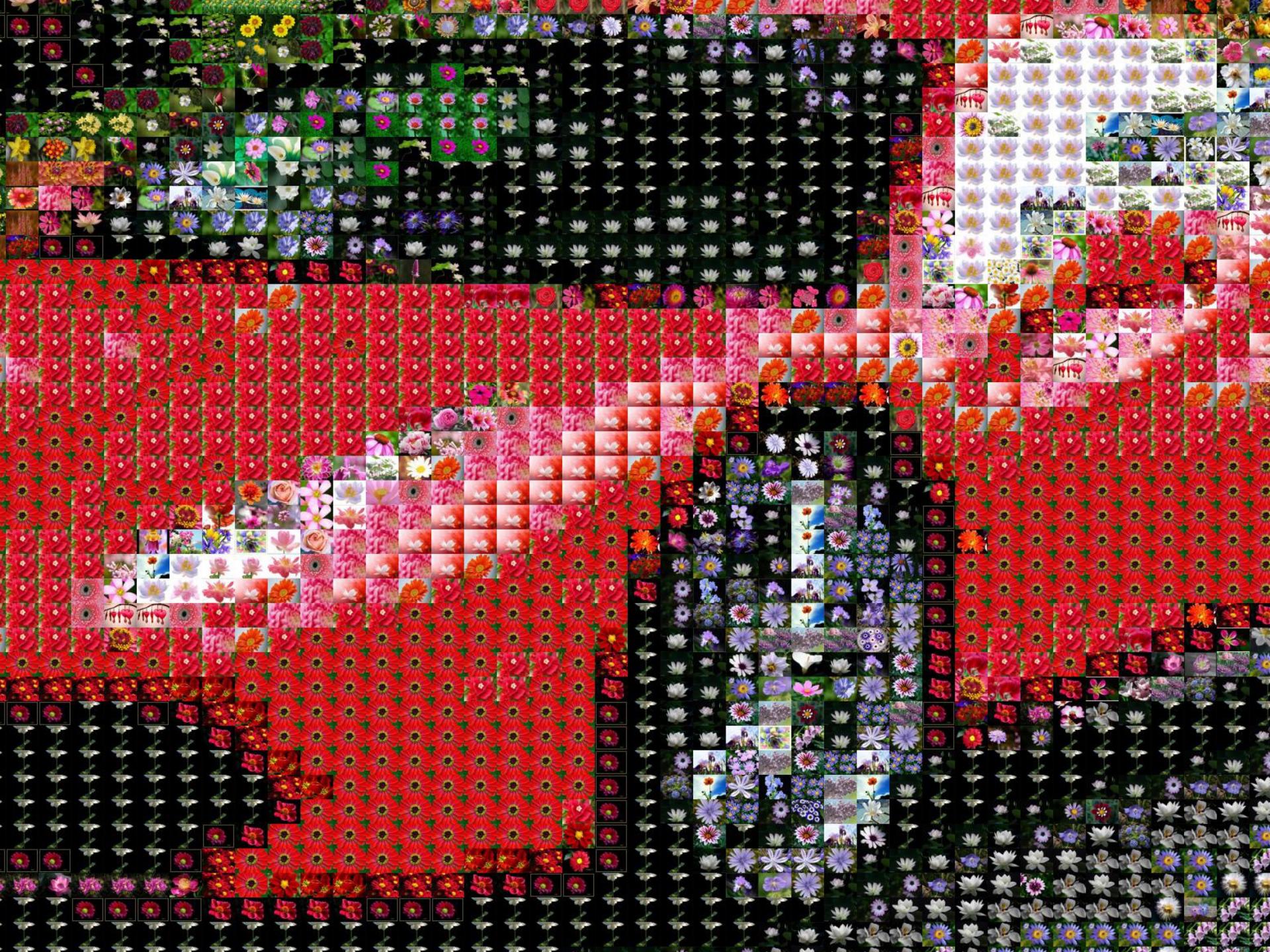
algoritm codat de voi

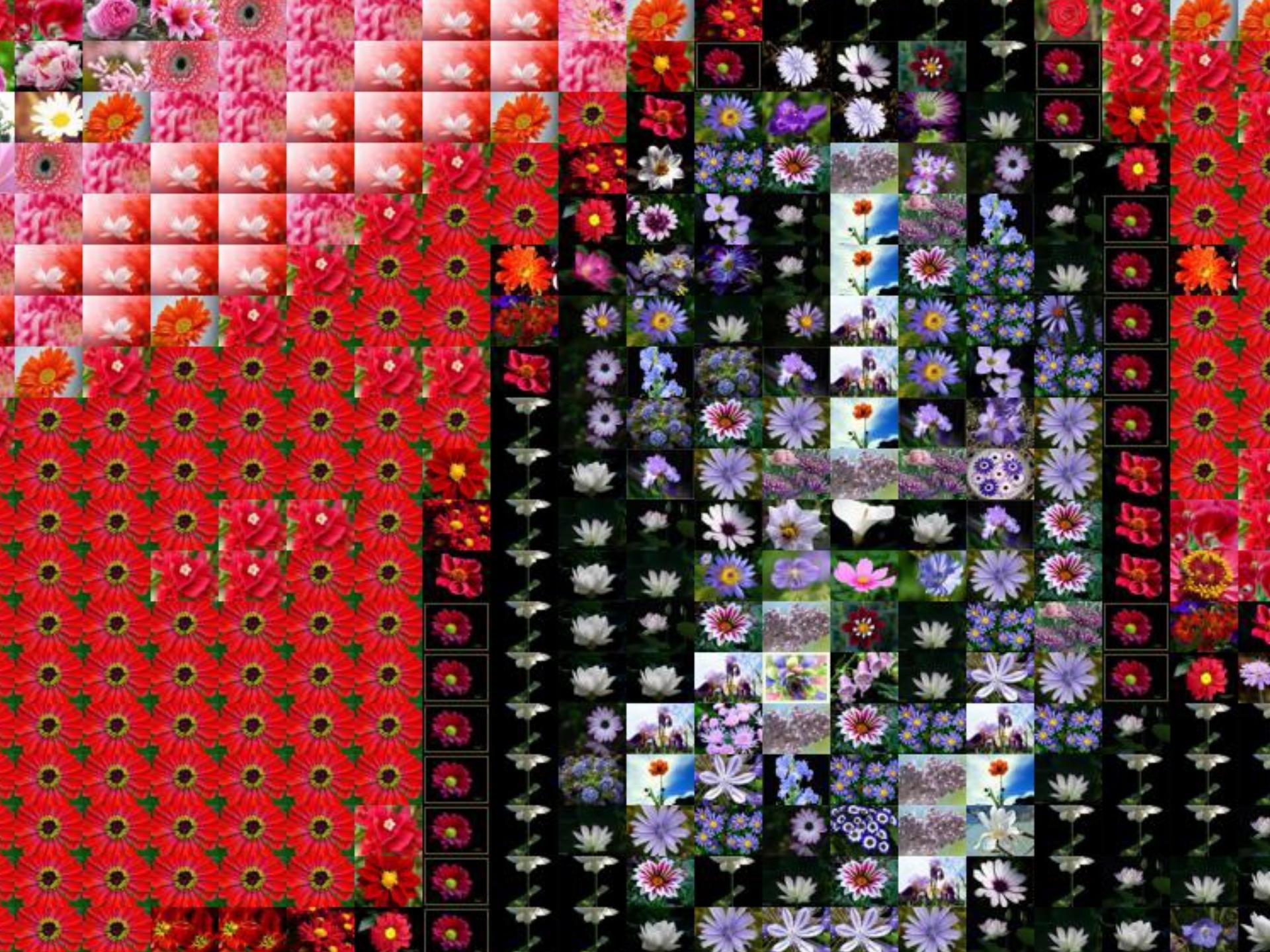


imagine mozaic









Algoritm

Input:

1. imagine de referință



2. colecție de imagini (piese)
500 de piese a 28 x 40 pixeli



3. număr piese în lățime ale
mozaicului (implicit = 100)

1	2		100
---	---	--	-----

Algoritm – varianta 1

Pași:

1. determină înălțimea mozaicului:
 - păstrează proporția (aspect ratio) imaginii de referință inițiale
 - multiplu de înălțimea unei piese
1. adaugă piese în mozaic pe un grid (caroaj): aplică un criteriu pentru a selecta piesa care se potrivește cel mai bine într-o poziție:
 - aleator (alegem piesele întâmplător)
 - culoarea medie cea mai apropiată (cea mai mică distanță euclidiană)

1	2		100
x			



Criterii

- aleator: alegem piesele la întâmplare, considerând că orice piesă la fel de bună ca cealaltă în orice poziție

imagine de referință



imagine mozaic



Criterii

- culoarea medie cea mai apropiată:
 - calculăm culoarea medie a pixelilor din blocul de înlocuit din imaginea de referință redimensionată;
 - culoarea medie este un triplet (r_{medie} , g_{medie} , b_{medie}) pentru imagini color sau un scalar i_{medie} pentru imagini gri;
 - calculăm culoarea medie a pixelilor din fiecare piesă;
 - calculăm cea mai apropiată (distanța euclidiană cea mai mică) culoare medie a unei piese de culoarea medie a blocului;
 - înlocuim blocul cu piesa aleasă.

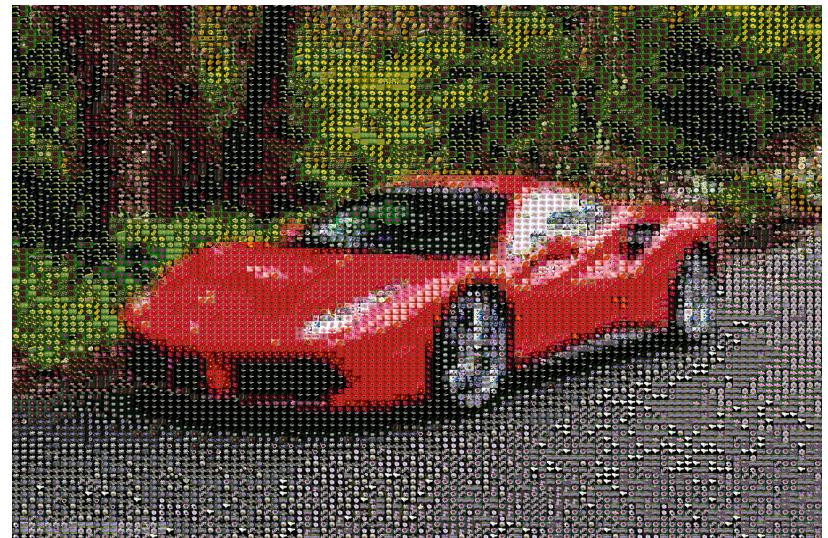
Criterii

- culoarea medie cea mai apropiată

imagine de referință



imagine mozaic



Criterii

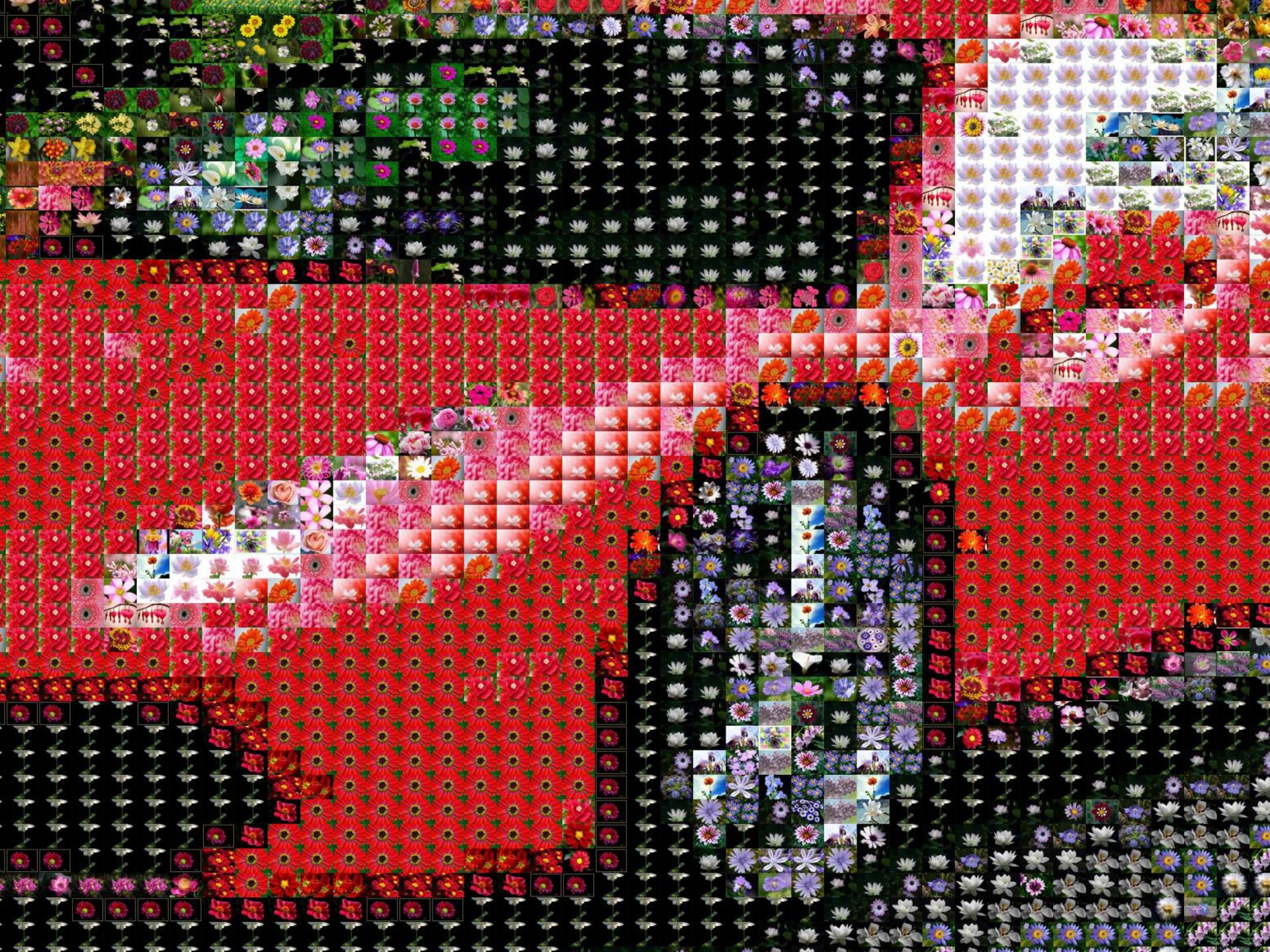
- culoarea medie cea mai apropiată

imagine de referință

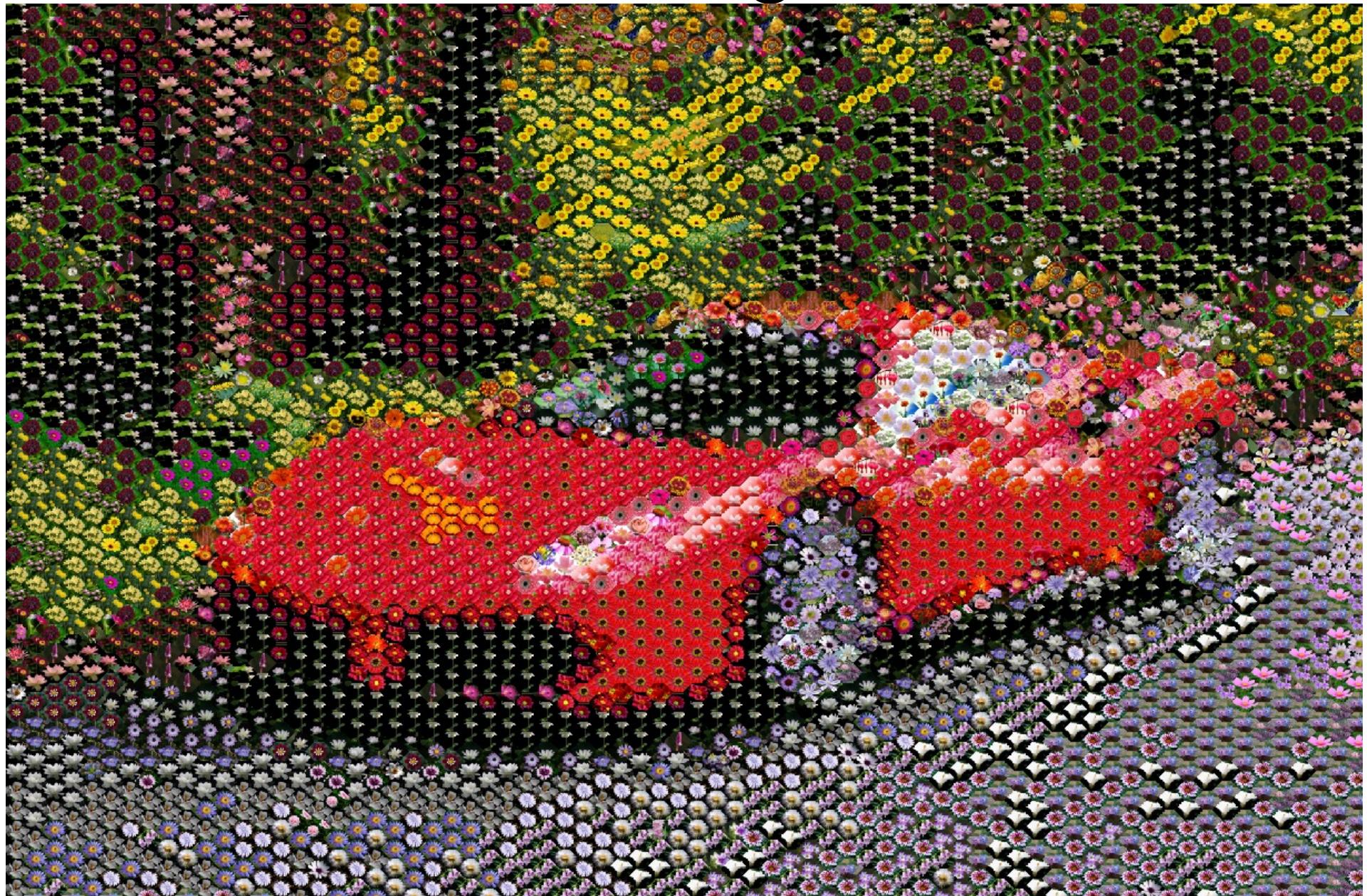


imagine mozaic

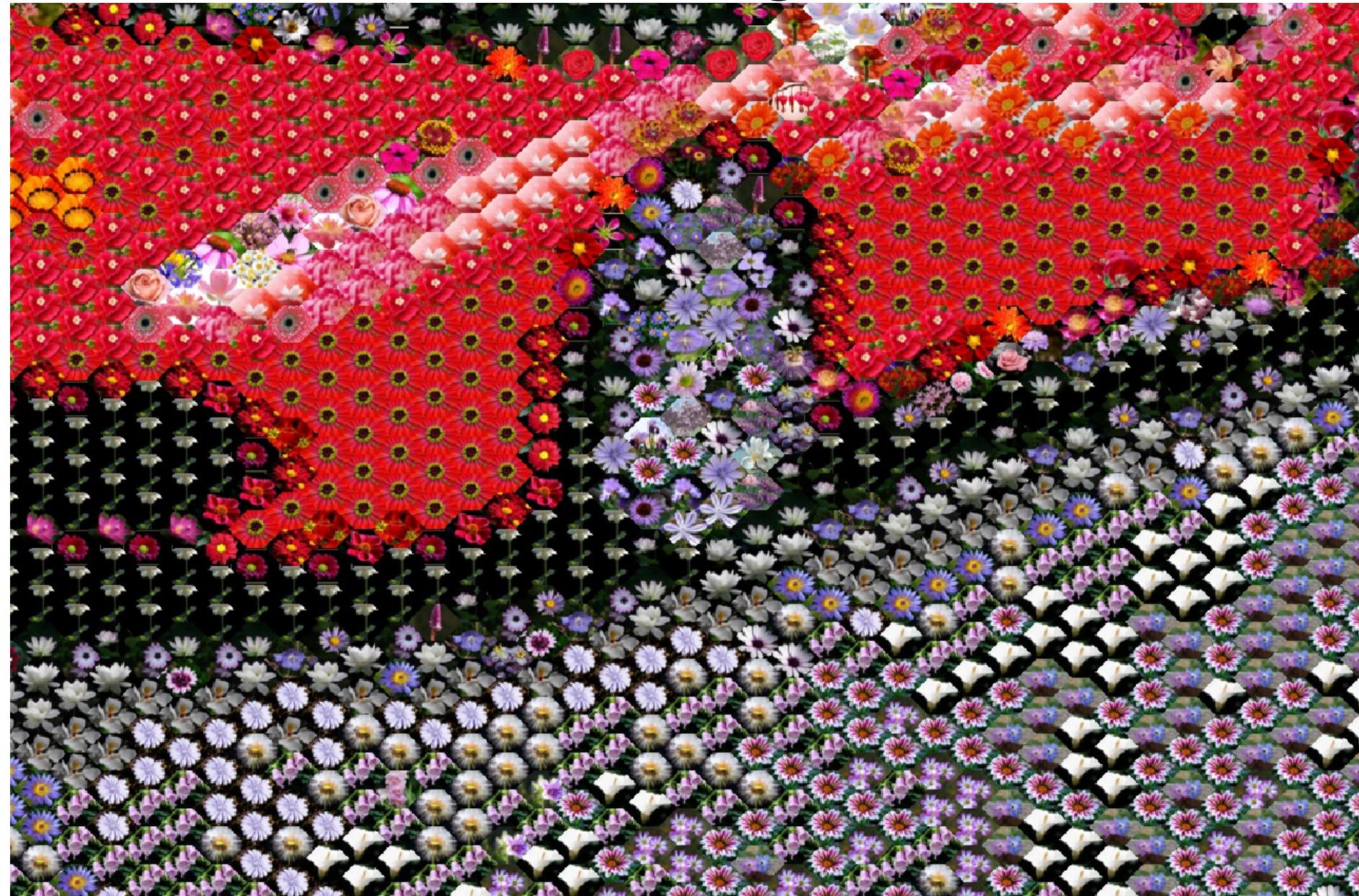




Piese hexagonale



Piese hexagonale



Piese hexagonale



Piese hexagonale



✗



≡



Piesă inițială

Mască (valori de 0 și 1)

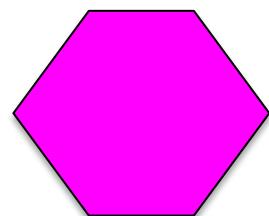
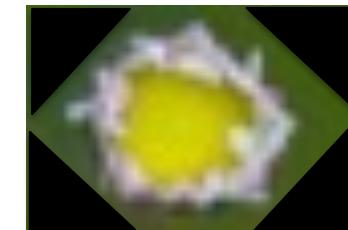
Piesă hexagonală



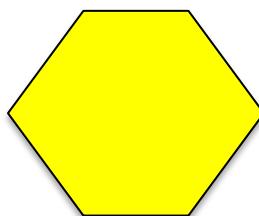
✗



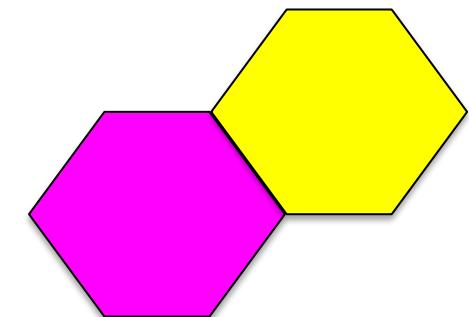
≡



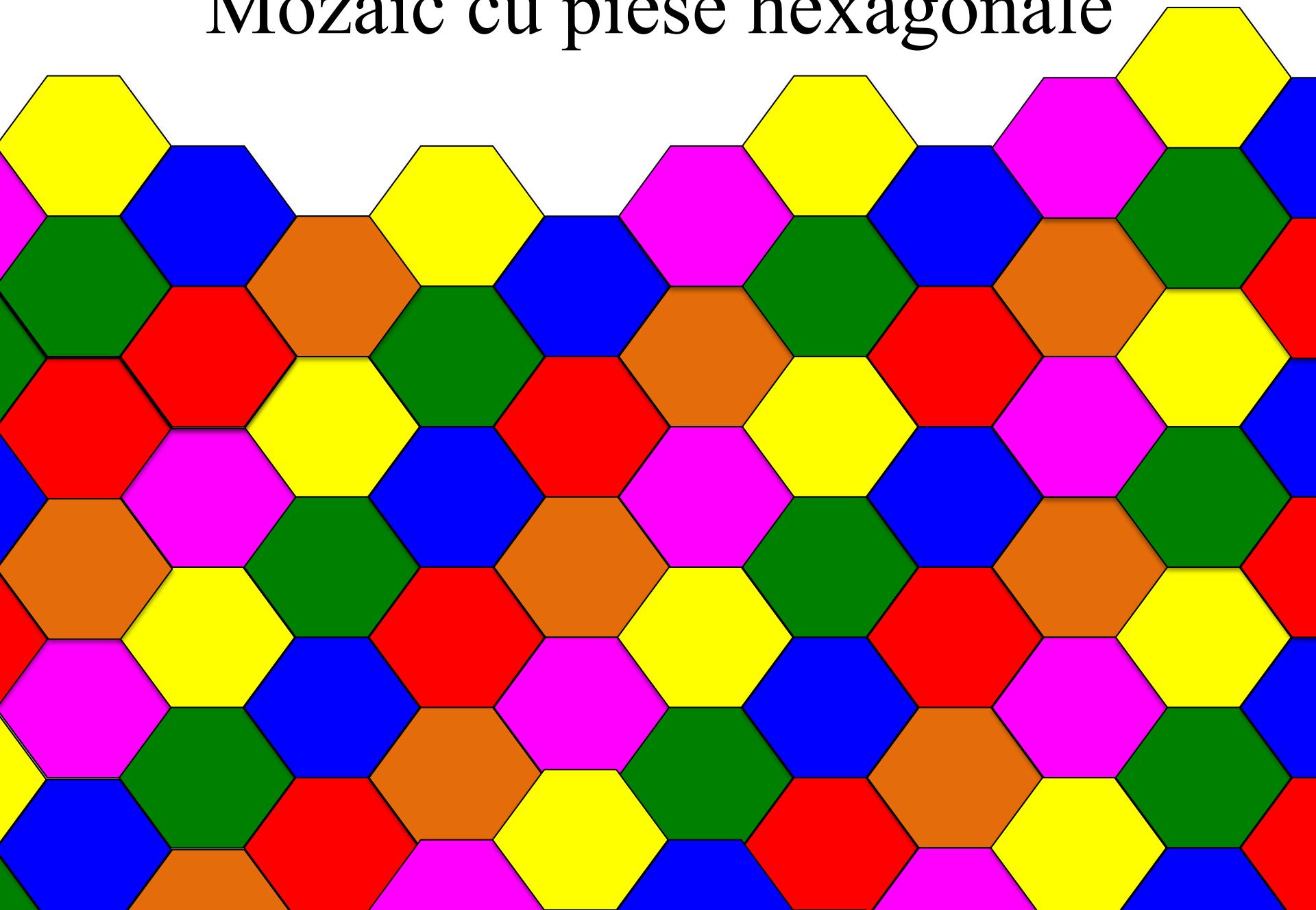
+



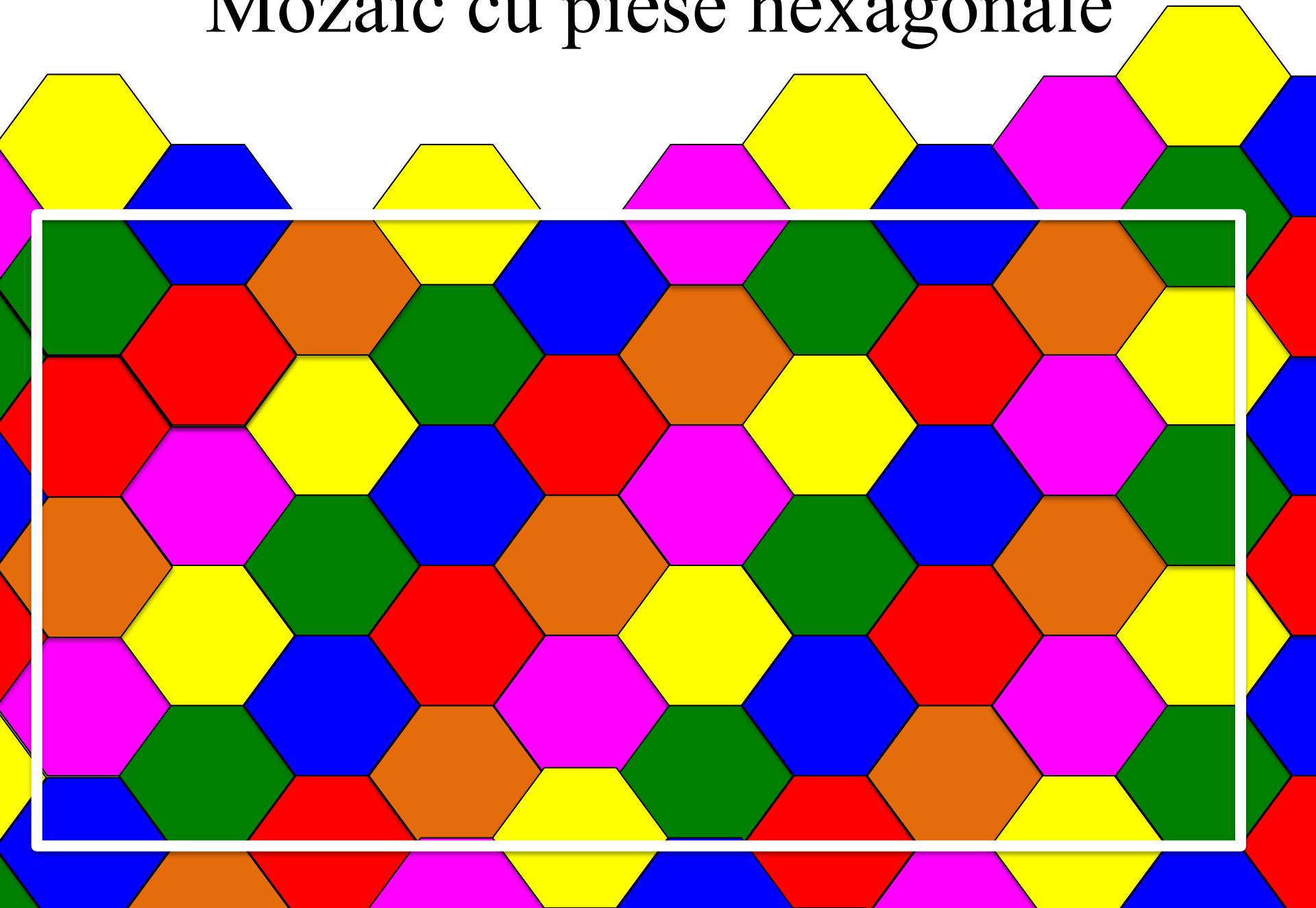
≡



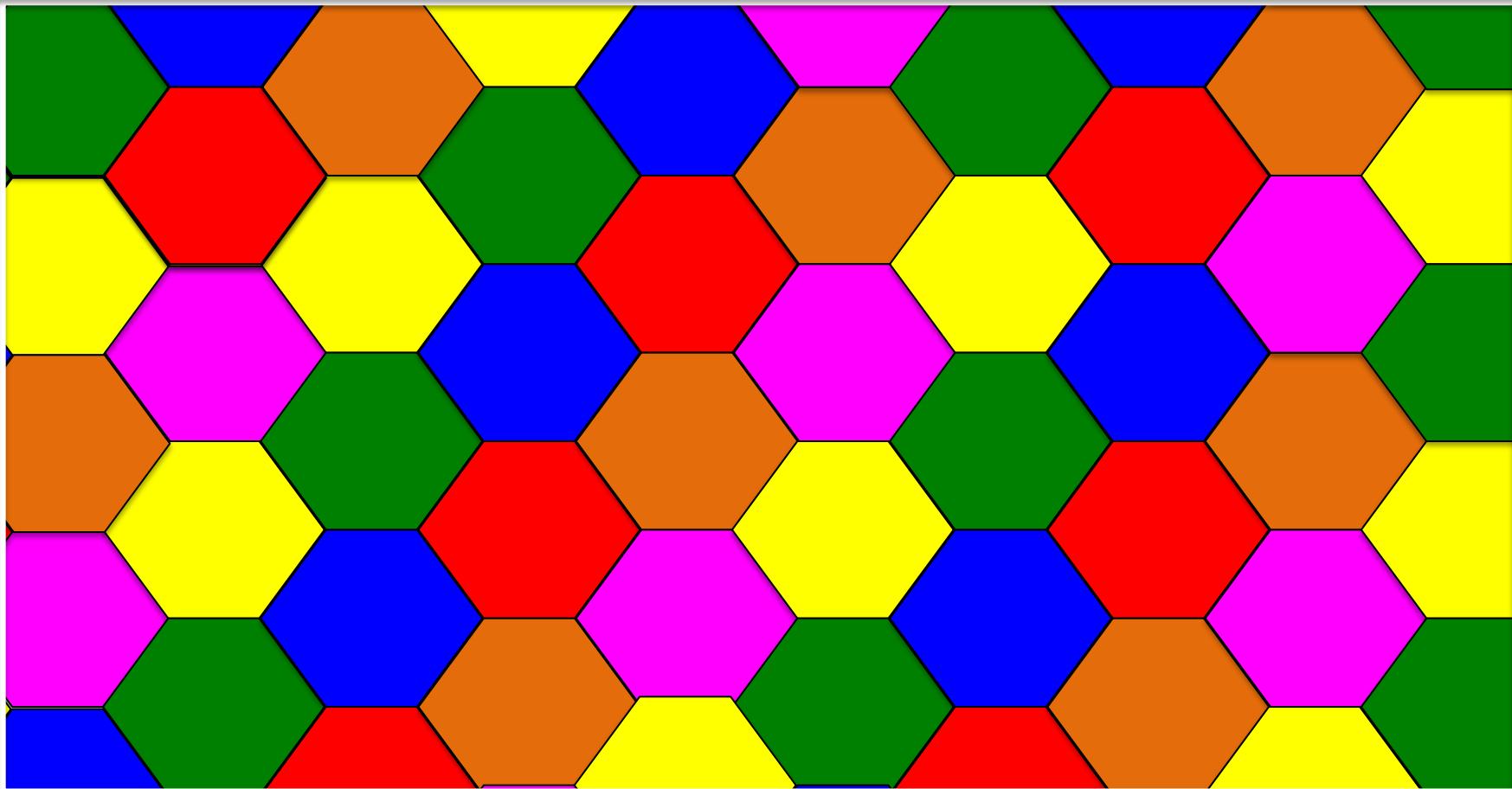
Mozaic cu piese hexagonale



Mozaic cu piese hexagonale



Mozaic cu piese hexagonale



Algoritm – varianta 2

Pași:

1. determină înălțimea mozaicului:
 - păstrează proporția (aspect ratio) imaginii de referință inițiale
 - multiplu de înălțimea unei piese
1. adaugă piese în mozaic în poziții aleatoare: aplică un criteriu pentru a selecta piesa care se potrivește cel mai bine într-o pozitie:
 - aleator (alegem piesele întâmplător)
 - culoarea medie cea mai apropiată (cea mai mică distanță euclidiană)

1	2		100

Algoritm – varianta 2

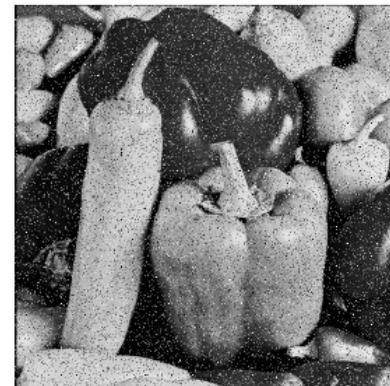
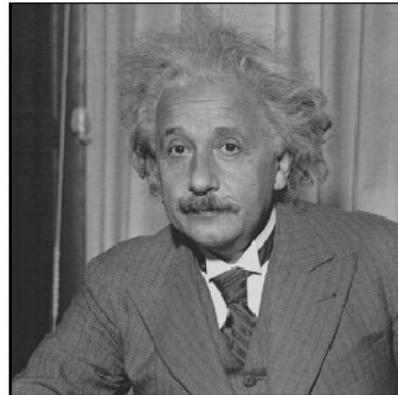


Algoritm – varianta 2

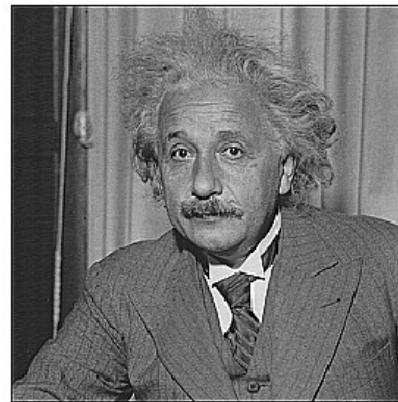


Filtrarea imaginilor

Exemplu de filtrare



Filtru de blurare

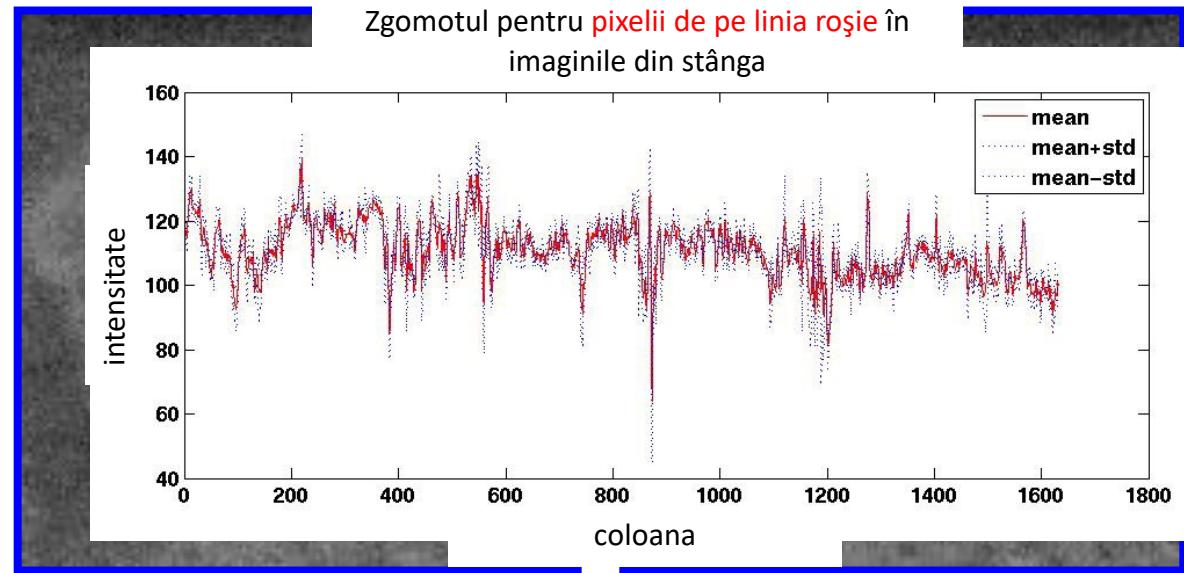
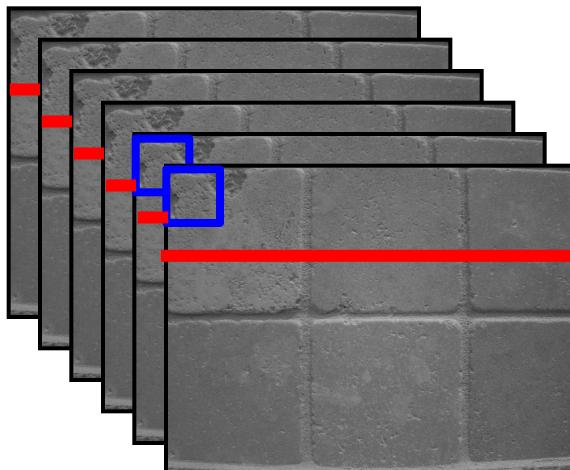


Filtru de accentuare



Filtru pentru eliminarea /
reducerea zgomotului

Motivătie: reducerea de zgomot



- Imperfecțiuni tehnologice ale senzorilor de imagine

$$I(x, y) = \hat{I}(x, y) + \eta(x, y)$$

imagine obținută imagine ideală zgomot

- Imagini multiple ale **aceleiasi scene statice** nu vor fi identice

Tipuri de zgromot



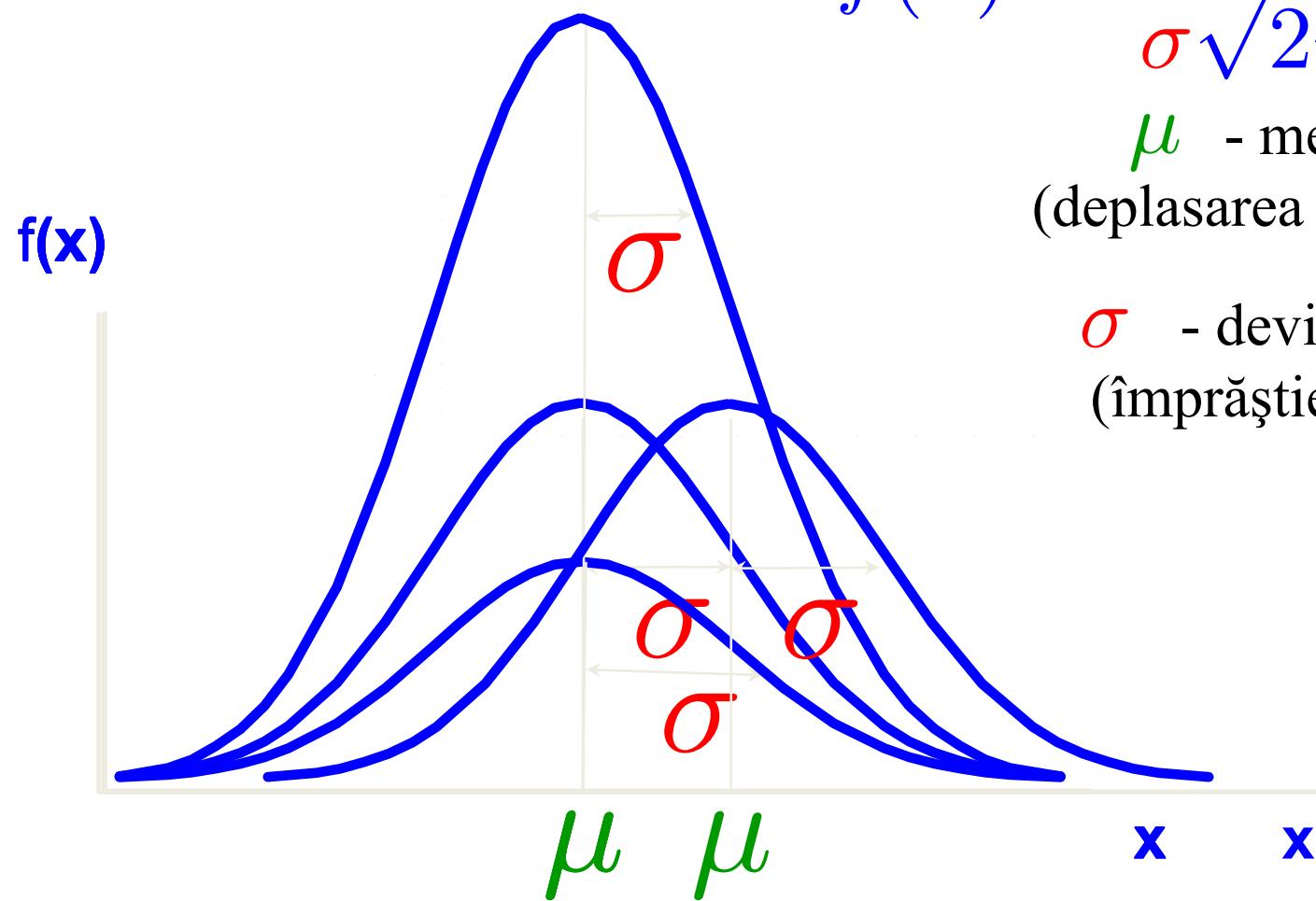
imagine originală

Distributia normală 1D (clopotul lui Gauss)

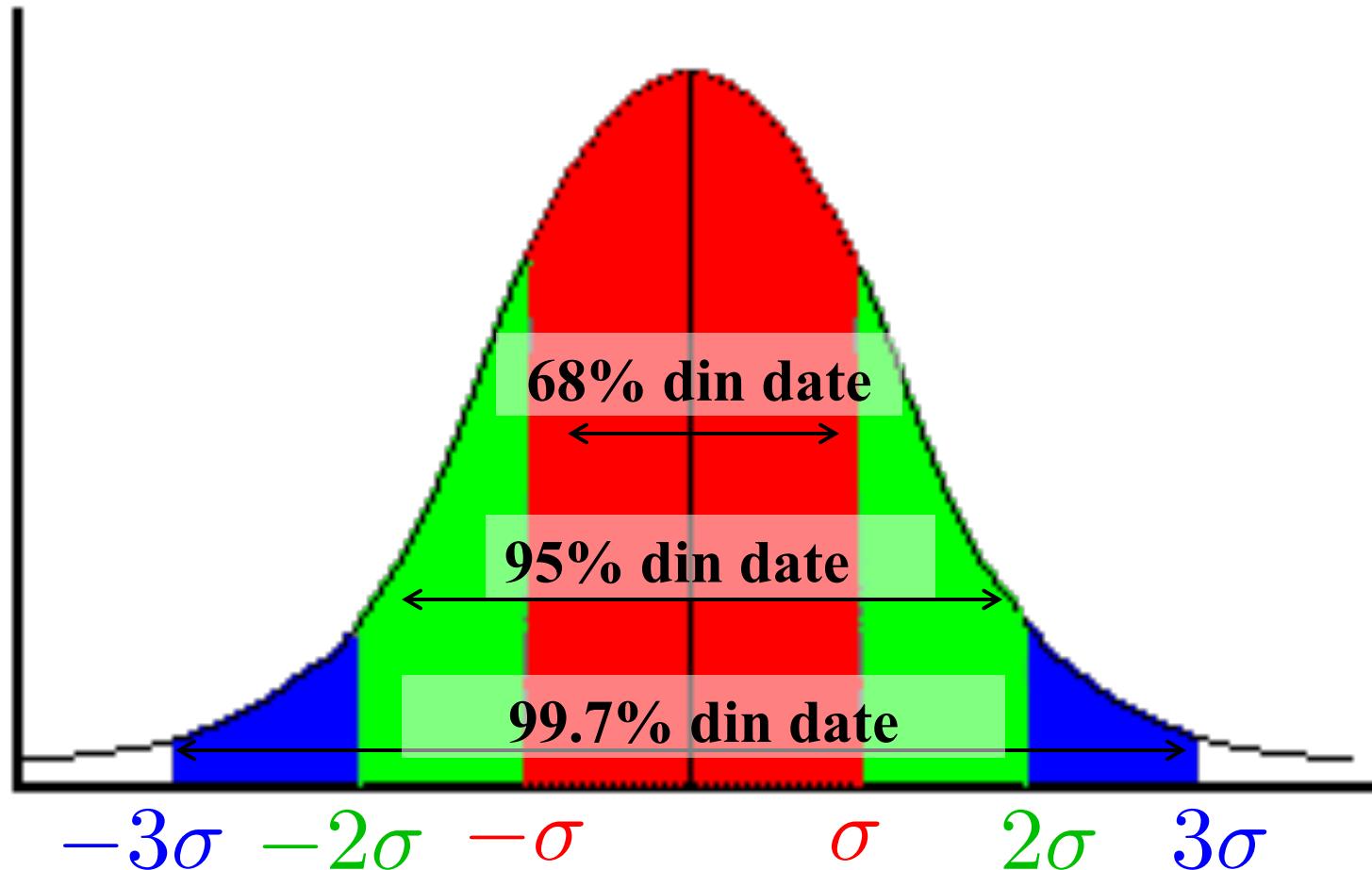
$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

μ - medie
(deplasarea distribuției)

σ - deviație standard
(împrăștierea distribuției)



Regula 68-95-99.7



Zgomot normal – efectul lui σ



sigma = 1

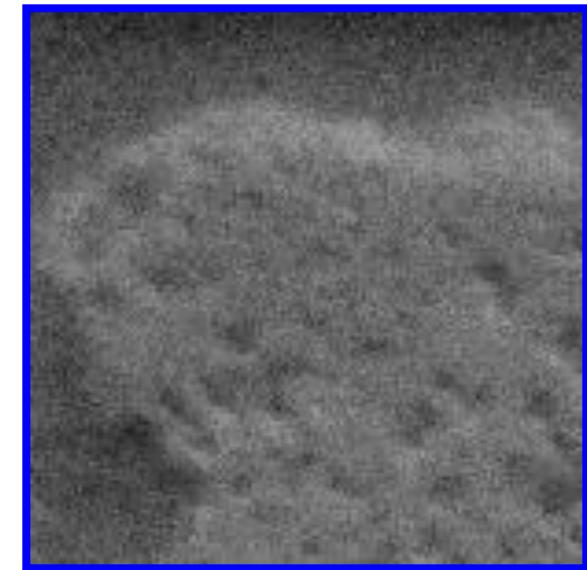
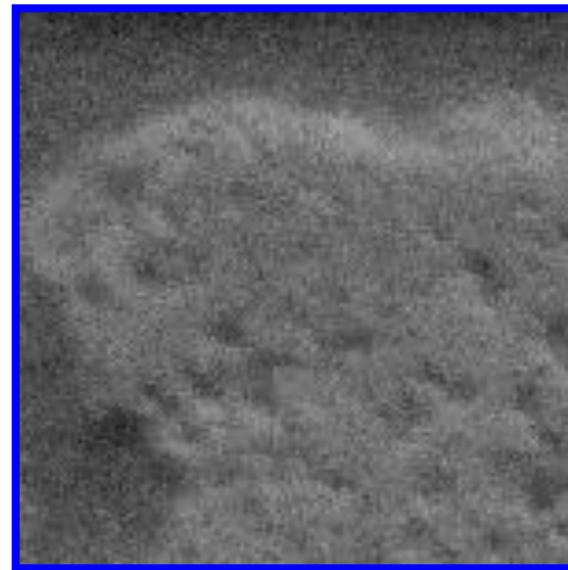
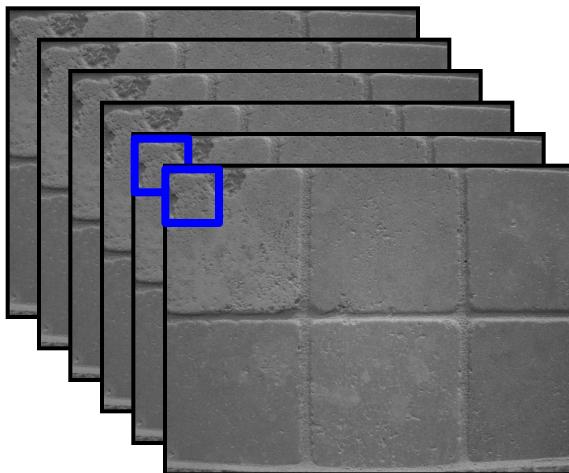


sigma = 5



sigma = 25

Motivație: reducerea de zgomot



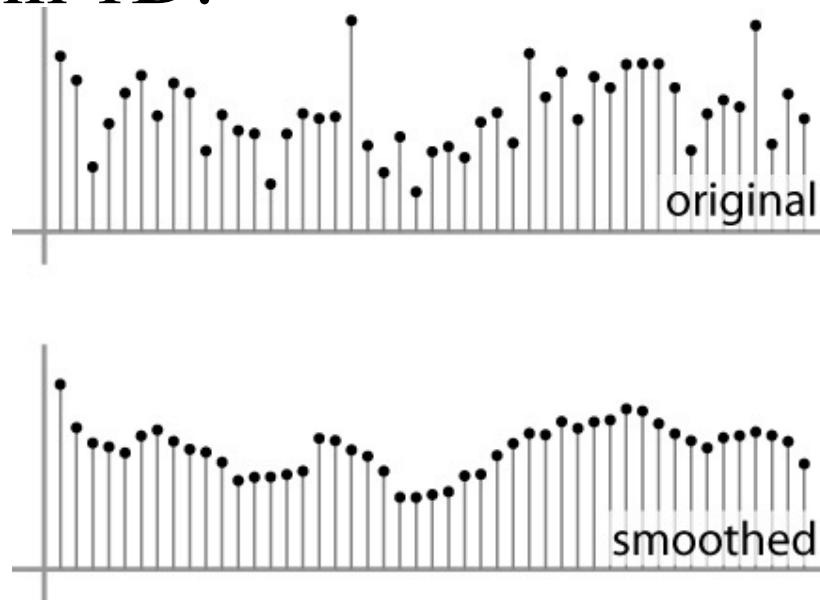
- Chiar și imagini multiple ale **aceleiași scene statice** nu vor fi identice.
- Cum putem reduce zgomotul, adică să estimăm adevăratele intensități?
- Dacă avem mai multe imagini luăm media
- **Dar dacă avem numai o singură imagine?**

O posibilă soluție

- Ipoteze:
 - pixelii “vecini” nu diferă prea mult între ei
 - zgomotul care afectează pixelii este independent
- Soluție:
 - înlocuim valoarea fiecărui pixel cu media valorilor pixelilor din “vecinătatea” lui

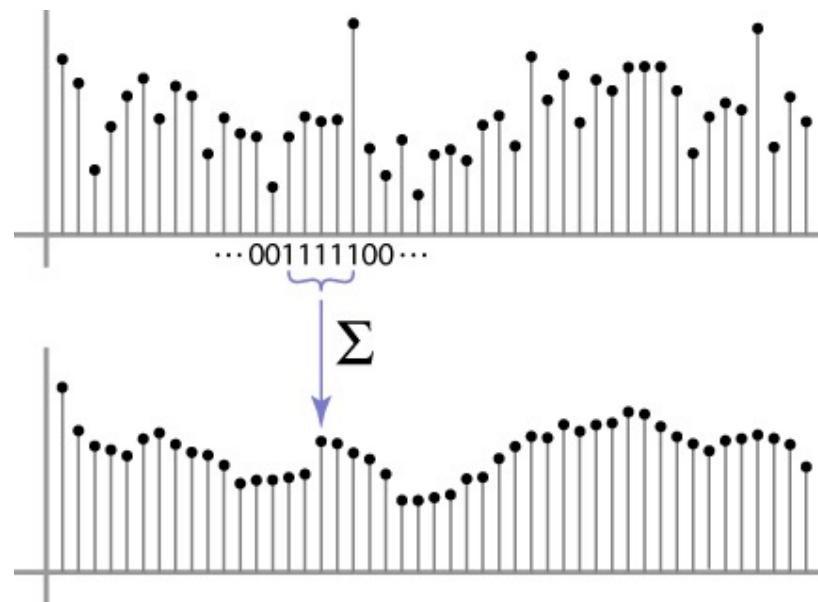
O posibilă soluție

- Înlocuim valoarea fiecărui pixel cu media valorilor pixelilor din vecinătatea lui
- Exemplu în 1D:



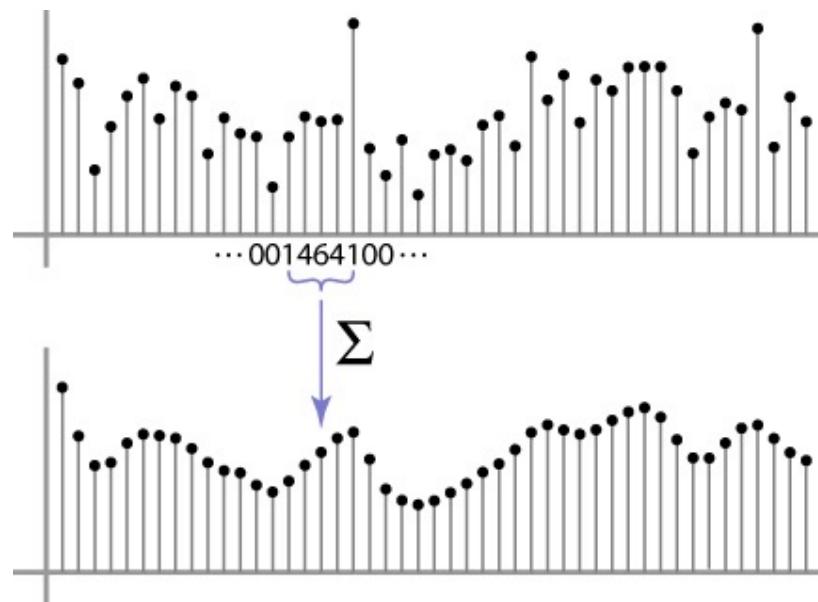
Medie ponderată

- Putem adaugă ponderi pentru fiecare pixel
- $Ponderi [1, 1, 1, 1, 1] / 5$



Medie ponderată

- Ponderi ne-uniforme $[1, 4, 6, 4, 1] / 16$



Medie în 2D

$$I[x, y]$$

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$$O[x, y]$$

			0							

Medie în 2D

$I[x, y]$

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$O[x, y]$

			0	10						

Medie în 2D

$$I[x, y]$$

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	90	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	90	0	0
0	0	0	90	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$$O[x, y]$$

			0	10	20					

Medie în 2D

$$I[x, y]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$O[x, y]$$

Medie în 2D

$$I[x, y]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$O[x, y]$$

Medie în 2D

$I[x, y]$

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$O[x, y]$

	0	10	20	30	30					

Medie în 2D

$$I[x, y]$$

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$$O[x, y]$$

	0	10	20	30	30					

Medie în 2D

 $I[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

 $O[x, y]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Corelație

- Pentru o vecinătate de dimensiuni $2k+1 \times 2k+1$:

$$O[i, j] = \underbrace{\frac{1}{(2k+1)^2}}_{\text{pondere egală pentru fiecare pixel}} \sum_{u=-k}^k \sum_{v=-k}^k I[i+u, j+v]$$

toți pixelii din vecinătatea pixelului (i,j) din imaginea I

(i-k,j-k)				(i-k,j+k)
		(i,j)		
(i+k,j-k)				(i+k,j+k)

Vecinătate de dimensiuni $(2k+1) \times (2k+1)$ centrată în (i,j)

Corelație

- Pentru o vecinătate de dimensiuni $2k+1 \times 2k+1$:

$$O[i, j] = \underbrace{\frac{1}{(2k+1)^2}}_{\begin{array}{l} \text{pondere egală} \\ \text{pentru fiecare pixel} \end{array}} \sum_{u=-k}^k \sum_{v=-k}^k I[i+u, j+v]$$

toți pixelii din vecinătatea pixelului (i,j) din imaginea I

- Generalizăm pentru a permite ponderi diferite:

$$O[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k \underbrace{F[u, v]}_{\text{ponderi ne-uniforme}} I[i+u, j+v]$$

Corelație

$$O[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k F[u, v] I[i + u, j + v]$$

Corelație: $O = F \otimes I$

(-k,-k)				(-k,+k)
		(0,0)		
(+k,-k)				(+k,+k)

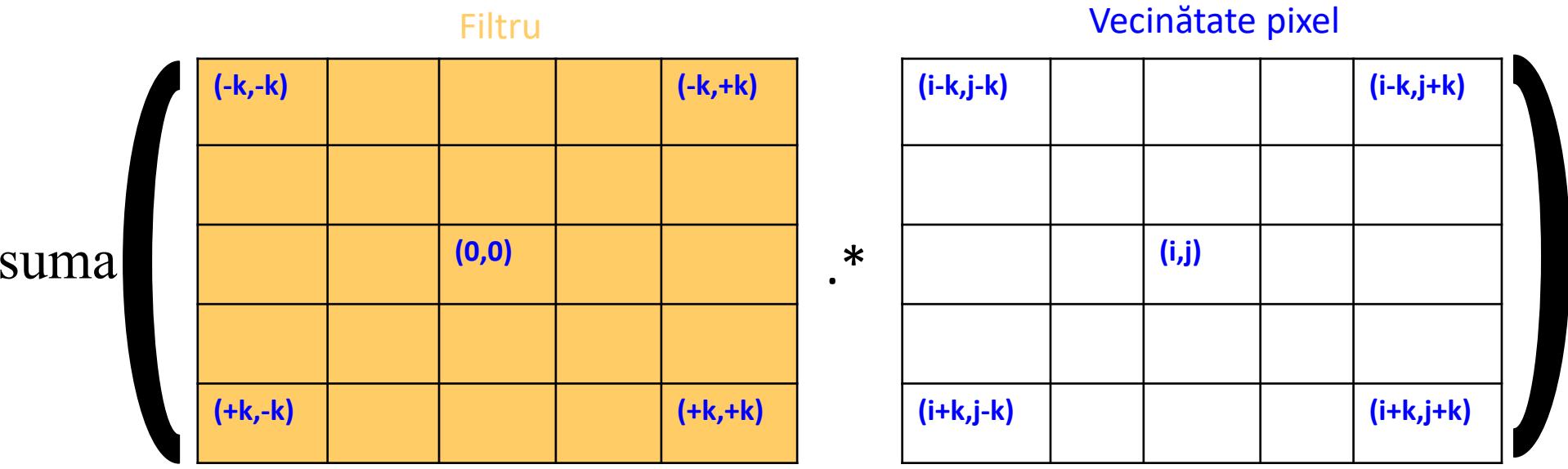
Filtrul F de dimensiuni $(2k + 1) \times (2k+1)$ centrat în $(0,0)$

Corelație

$$O[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k F[u, v] I[i + u, j + v]$$

Corelație: $O = F \otimes I$

Filtrarea (liniară a) unei imagini: înlocuim fiecare pixel cu o combinație (liniară) a “vecinilor” săi.



Corelație

$$O[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k F[u, v] I[i + u, j + v]$$

Corelație: $O = F \otimes I$

Filtrarea (liniară a) unei imagini: înlocuim fiecare pixel cu o combinație (liniară) a “vecinilor” săi.

F se mai numește filtru, kernel, mască (de filtrare).

Filtrul F - conține ponderile pixelilor folosiți în combinația (liniară).

Filtrul de medie

- Ce ponderi are filtrul F pentru exemplul precedent (medie în 2D)?

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & ? & 1 \\ 1 & 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 0 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 90 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

filtru de medie
(filtru pătrat)

$$F[u, v] \quad \otimes \quad I[x, y] \quad O[x, y]$$

0	10	20	30	30					

$$O = F \otimes I$$

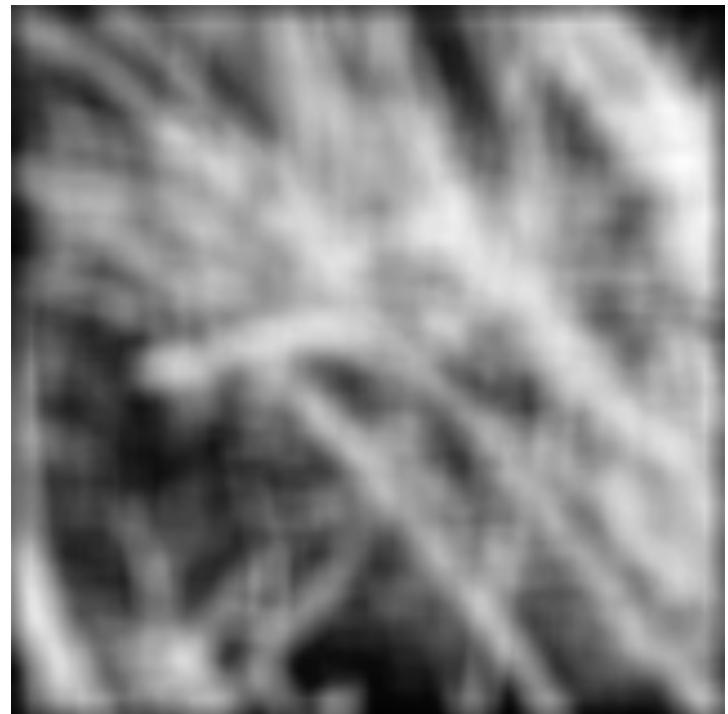
Blurarea (netezirea) unei imagini cu un filtru de medie



pictograma pentru filtru de medie:
alb = valoare mare, negru = valoare mică



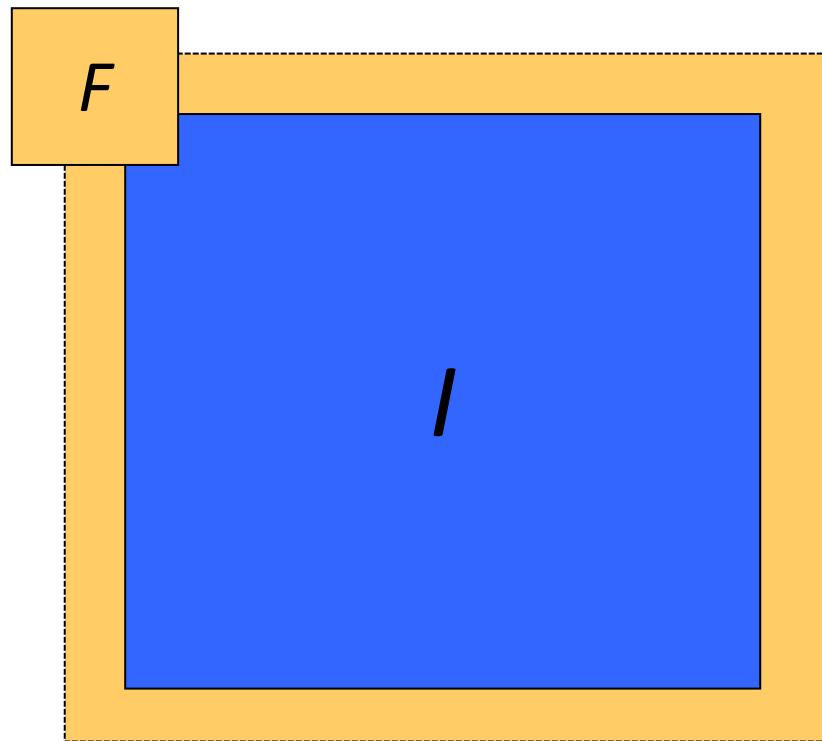
Imagine inițială



Imagine filtrată

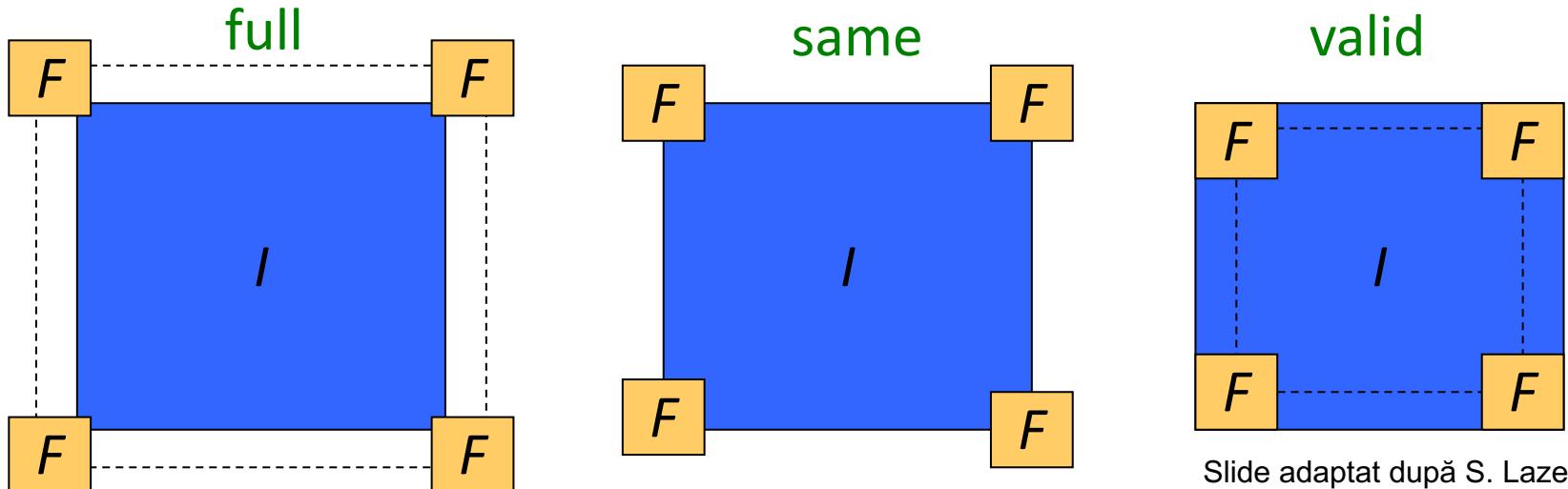
Dimensiune output

- Care este dimensiunea imaginii filtrate?



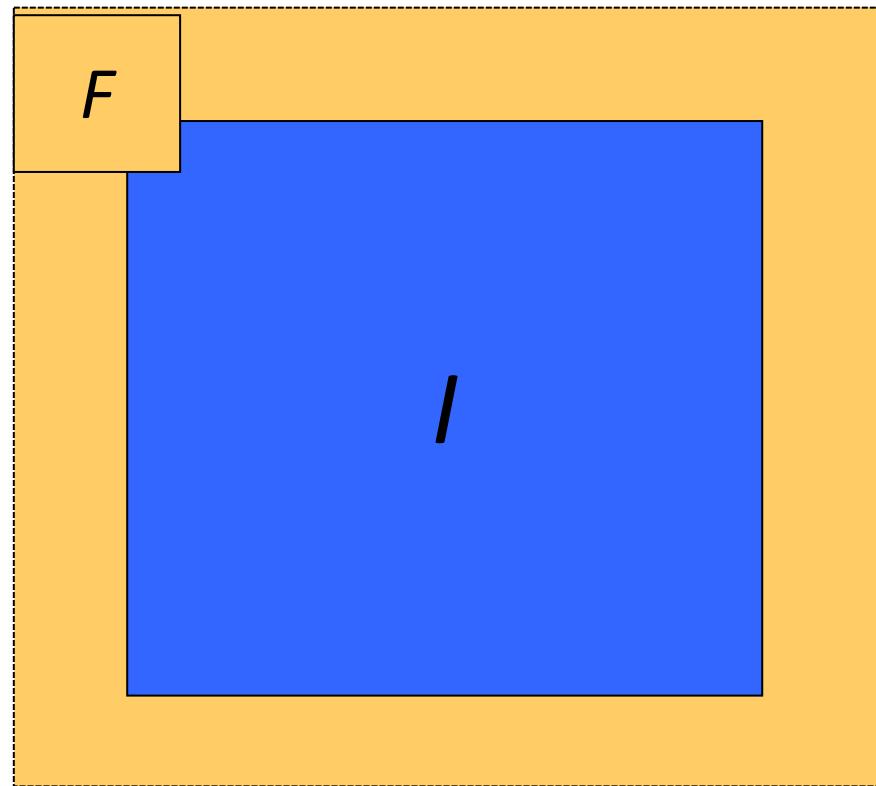
Dimensiune output

- Care este dimensiunea imaginii filtrate?
- PYTHON: `scipy.signal.convolve2d`
- opțiuni pentru filtrare
 - ‘full’: $\text{dim}(O) = \text{dim}(I) + \text{dim}(F)$
 - ‘same’: $\text{dim}(O) = \text{dim}(I)$
 - ‘valid’: $\text{dim}(O) = \text{dim}(I) - \text{dim}(F)$



Pixelii din afara imaginii

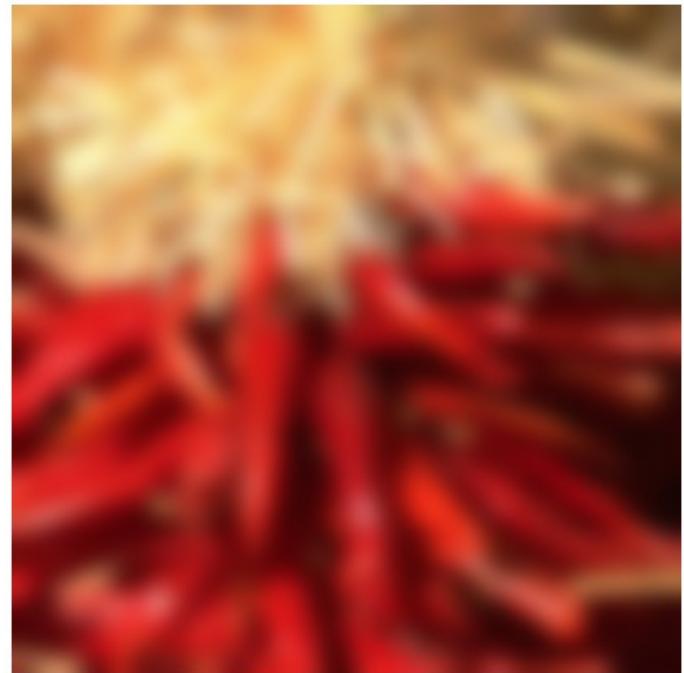
- Ce valori au pixelii din afara imaginii ?



- Fereastra filtrului depăseste marginea imaginii

Pixelii din afara imaginii

- Ce valori au pixelii din afara imaginii ?
 - extrapolăm valori
 1. pixeli = 0 (negru)
 2. copiază pixelii în mod circular
 3. copiază pixelii de la marginea imaginii
 4. oglindește pixelii de la marginea imaginii

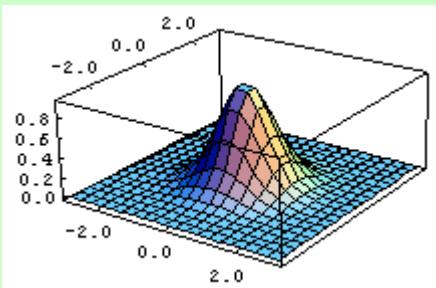


Filtru normal (Gaussian)

- Cei mai apropiati pixeli vecini au o pondere mai mare in imaginea filtrata

Acum filtrele aproximeaza distributia normala in 2D:

$$f(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



$$\frac{1}{16} \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix} F[u, v]$$

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	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

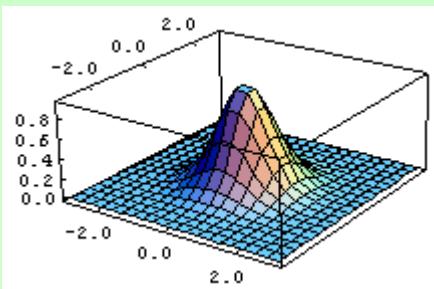
$$I[x, y]$$

Filtru normal (Gaussian)

- Cei mai apropiati pixeli vecini au o pondere mai mare in imaginea filtrata

Acum filtrele aproximeaza distributia normala in 2D:

$$f(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



$$\frac{1}{16} \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix} F[u, v]$$

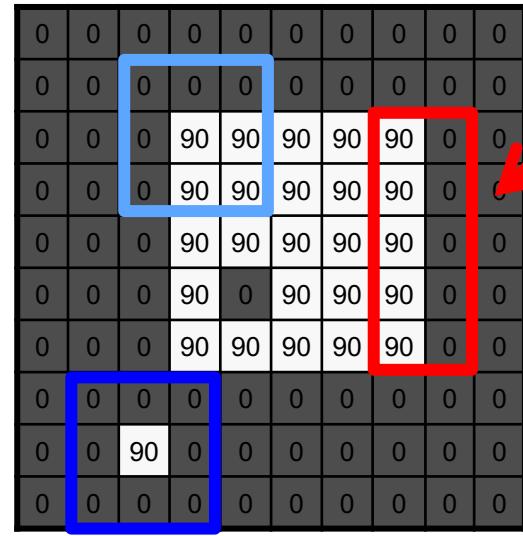
0	0	0	0	0	0	0	0	0	0	0
0	0	6	17	23	23	23	17	6	0	0
0	0	17	51	68	68	68	51	17	0	0
0	0	23	68	90	90	90	68	23	0	0
0	0	23	62	79	84	90	68	23	0	0
0	0	23	56	68	79	90	68	23	0	0
0	0	17	45	56	62	68	51	17	0	0
0	6	17	23	23	23	23	17	6	0	0
0	11	23	11	0	0	0	0	0	0	0
0	6	11	6	0	0	0	0	0	0	0

$$I[x, y]$$

Blurarea unei imagini

Filtru de medie

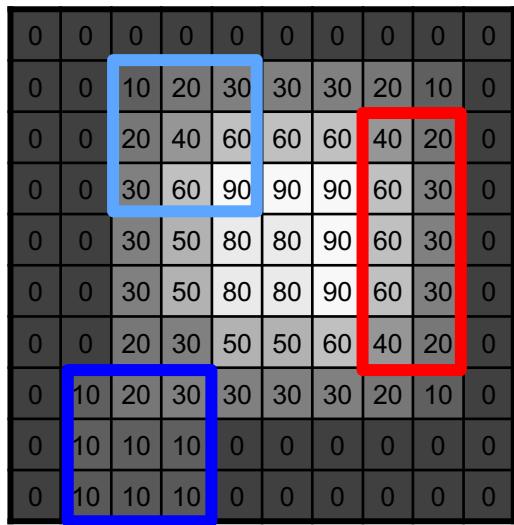
$$\frac{1}{9} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$



muchie
verticală

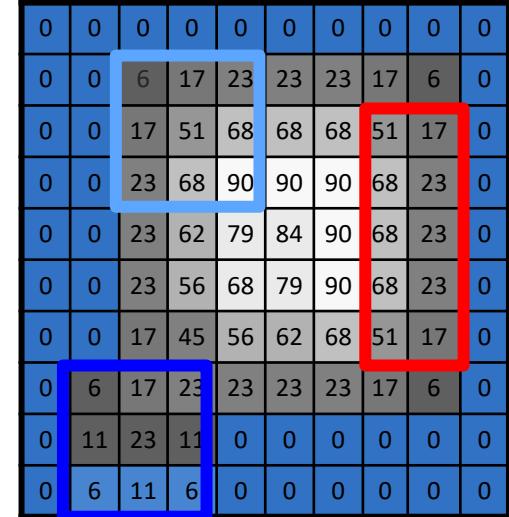
Filtru normal

$$\frac{1}{16} \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

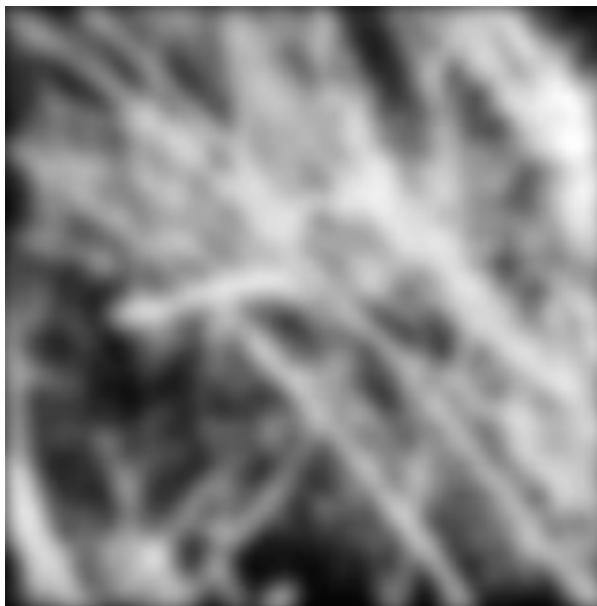


OBSERVAȚII:

1. se elimină regiunile din imagine cu schimbări brusă de intensitate (frecvențe înalte) - muchiile
2. se micșorează distanțele dintre pixeli cu intensitate mare și pixeli cu intensitate mică



Blurarea unei imagini



filtru Gaussian:
alb = valoare mare,
negru = valoare mică



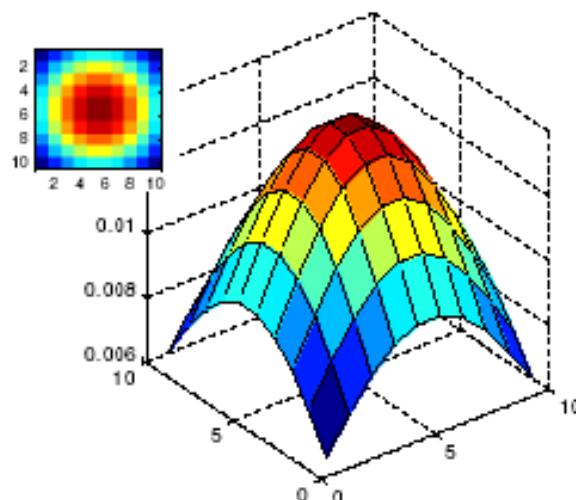
filtru de medie

Filtre normale (Gaussiene)

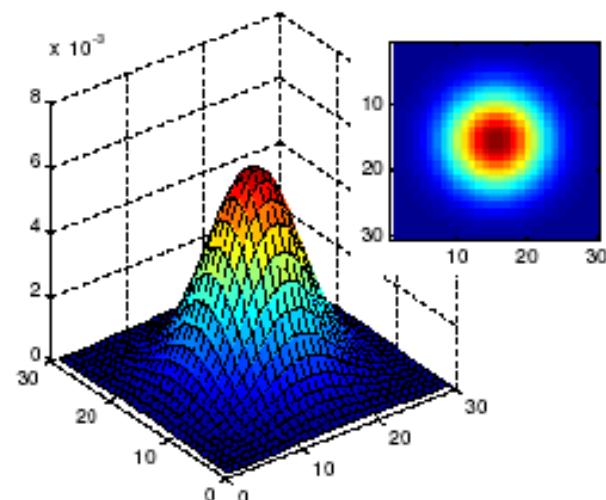
- Care sunt parametri ce definesc filtrul?

1. Dimensiunea

- funcțiile normale (Gaussiene) au suport infinit (> 0), însă filtrele au dimensiune finită



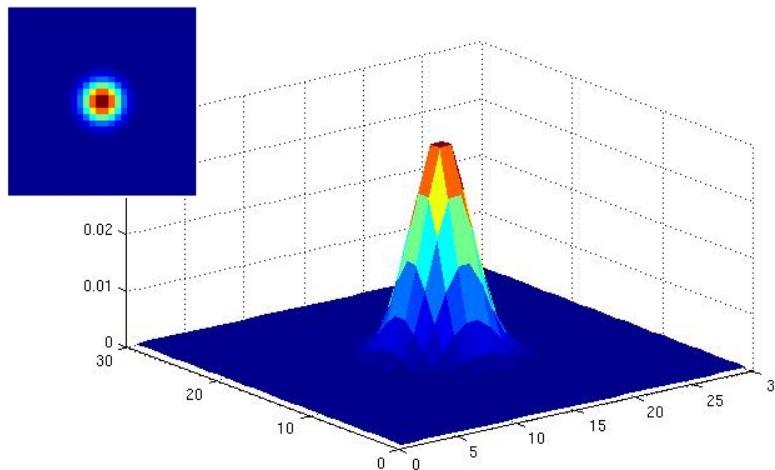
$\sigma = 5$
10 x 10



$\sigma = 5$
30 x 30

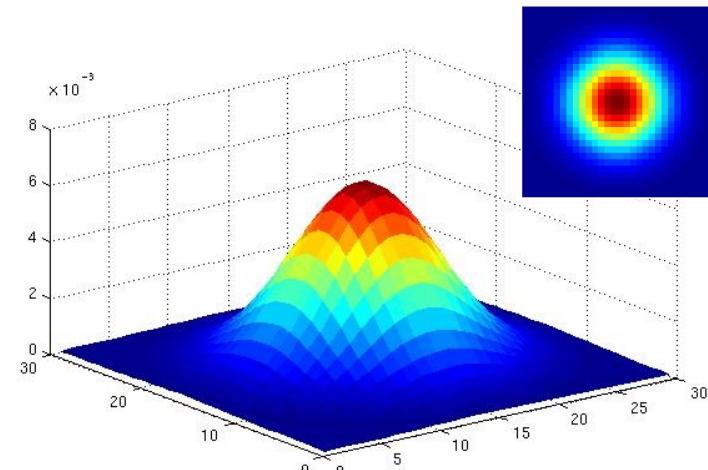
Filtre normale (Gaussiene)

- Care sunt parametri ce definesc filtrul?
- 2. Deviația standard:** cât de “neted” e filtrul



$\sigma = 2$

30 x 30

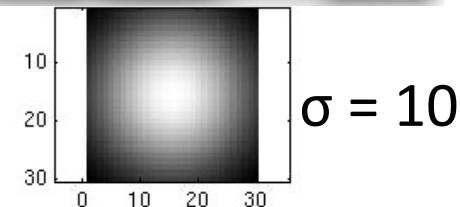
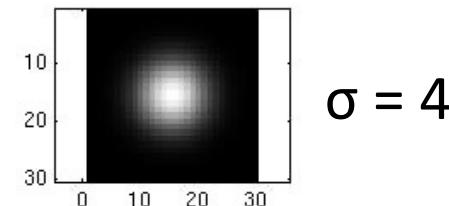
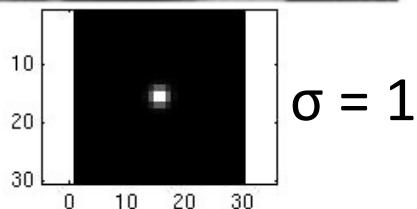


$\sigma = 5$

30 x 30

Blurarea unei imagini cu un filtru Gaussian

Parametrul σ – controlează “netezimea” filtrului



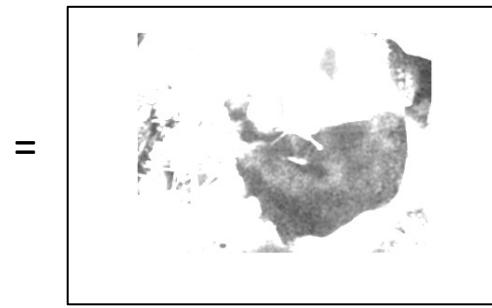
Proprietăți ale filtrelor de blurare ("smoothing")

1. valori pozitive;
1. suma lor = 1 → pentru regiuni constante (toți pixelii au aceeași valoare → perete alb), output = input;
1. gradul de blurare/netezire este proporțional cu dimensiunea filtrului;
1. elimină regiunile de pixeli cu varianță mare în intensitate (frecvențe înalte); se mai numesc filtre "trece-jos" ("low-pass")

Proprietăți ale filtrelor de blurare ("smoothing")

ce se întâmplă (ce caracteristici are imaginea rezultată) dacă avem un filtru de blurare care nu este normalizat (suma elementelor filtrului $\neq 1$)?

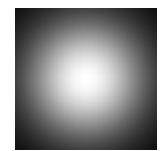
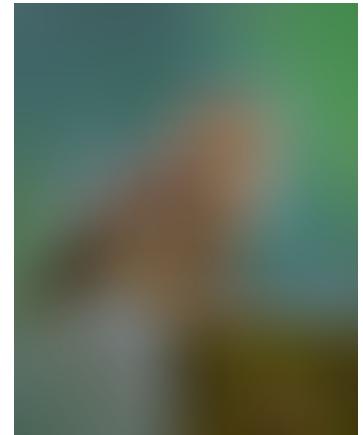
1	1	1
1	1	1
1	1	1



Dacă suma elementelor filtrului $> 1 \rightarrow$ imaginea se luminează
Dacă suma elementelor filtrului $< 1 \rightarrow$ imaginea se întunecă

Filtrarea imaginilor color

$$I(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix} \longrightarrow F \otimes I = \begin{bmatrix} F \otimes r \\ F \otimes g \\ F \otimes b \end{bmatrix}$$



$\sigma = 30$ pixels

Filtrarea imaginilor color

PYTHON: $f = \text{np.ones}((1,9))/9$
 $F = \text{cv2.filter2D(img,-1,f)}$



Imagine inițială I



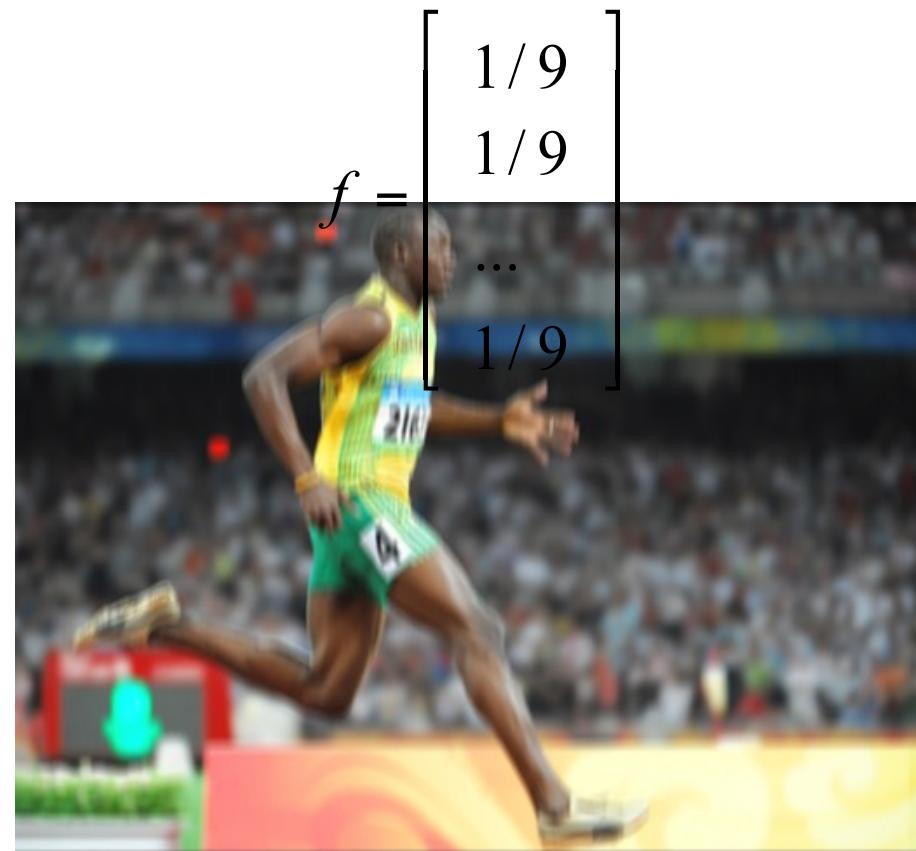
Imagine filtrată $f * I$

Filtrarea imaginilor color

PYTHON: $f = \text{np.ones}((9,1))/9$
 $F = \text{cv2.filter2D(img,-1,f)}$



Imagine inițială I



Imagine filtrată $f * I$

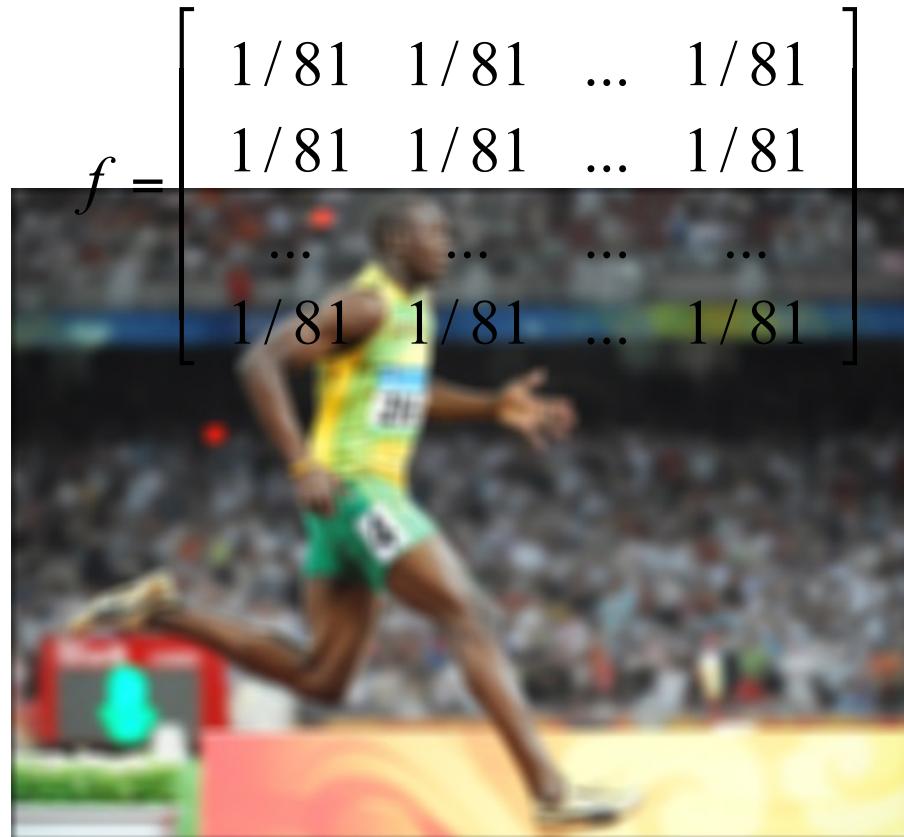
$$f = \begin{bmatrix} 1/9 \\ 1/9 \\ \dots \\ 1/9 \end{bmatrix}$$

Filtrarea imaginilor color

PYTHON: `f = np.ones((9,9))/81`
`F = cv2.filter2D(img,-1,f)`



Imagine inițială I



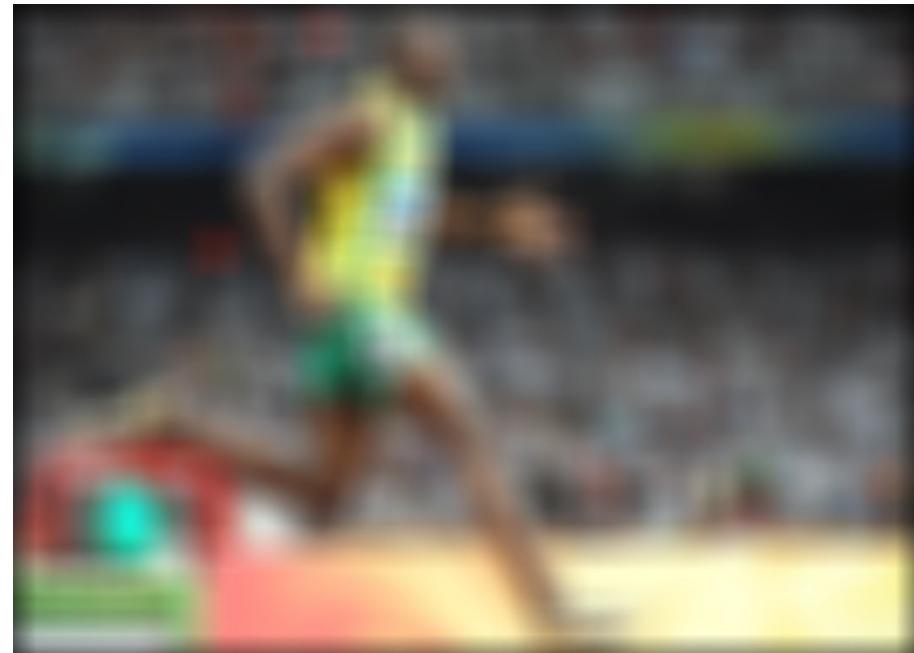
Imagine filtrată $f * I$

Filtrarea imaginilor color

PYTHON: $f = \text{np.ones}((27,27))/(27*27)$
 $F = \text{cv2.filter2D(img, -1, f)}$



Imagine inițială I



Imagine filtrată $f * I$

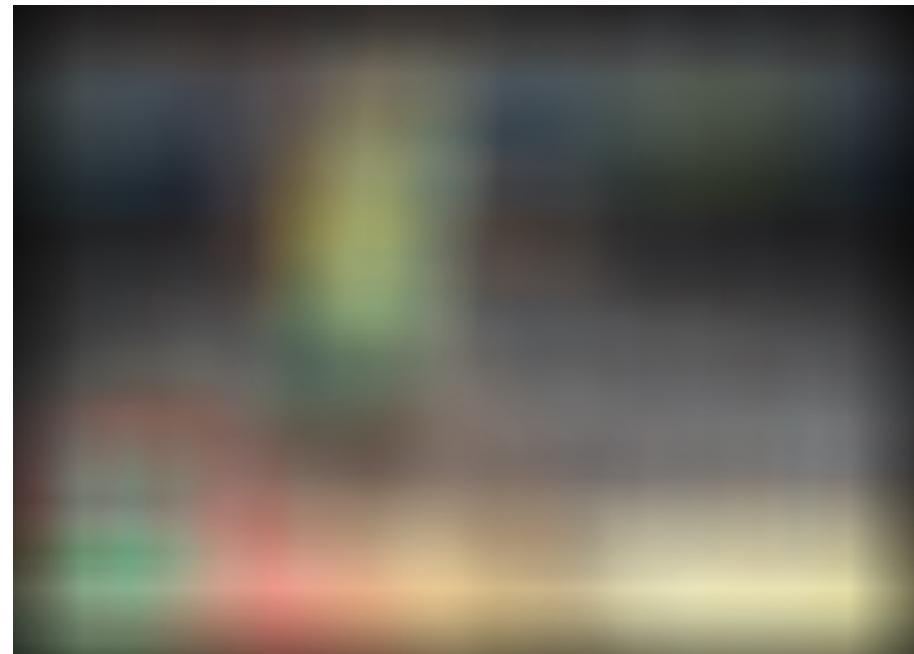
Filtrarea imaginilor color

PYTHON: $f = \text{np.ones}((81,81))/(81*81)$

$F = \text{cv2.filter2D}(\text{img}, -1, f)$



Imagine inițială I



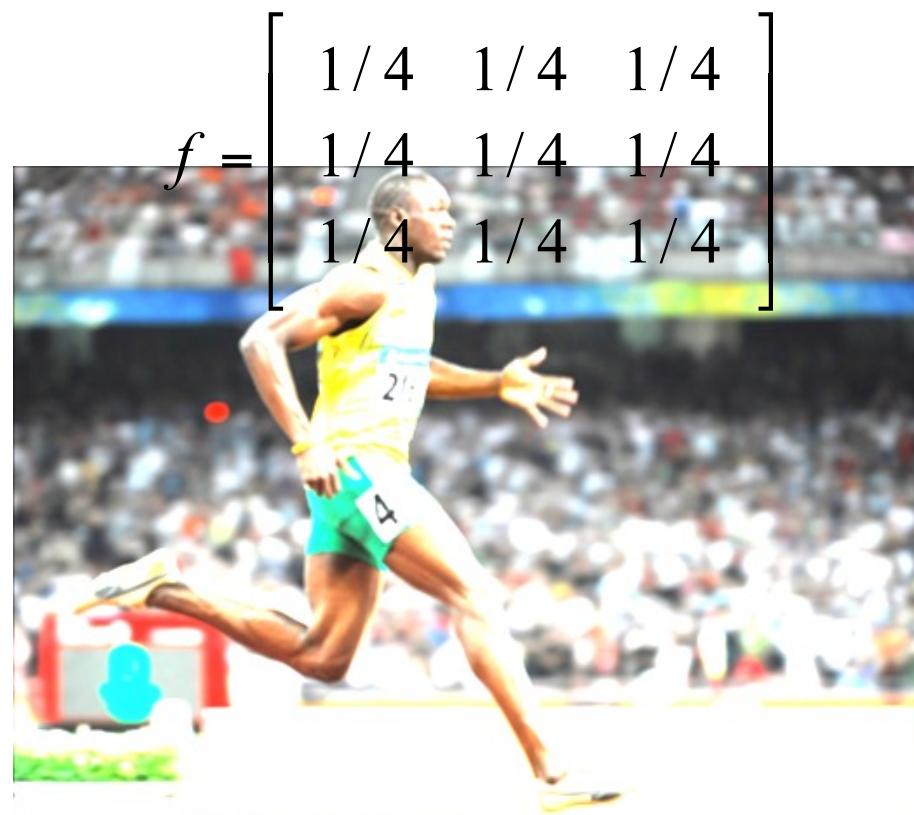
Imagine filtrată $f * I$

Filtrarea imaginilor color

PYTHON: $f = \text{np.ones}((3,3))/4$
 $F = \text{cv2.filter2D(img,-1,f)}$



Imagine inițială I



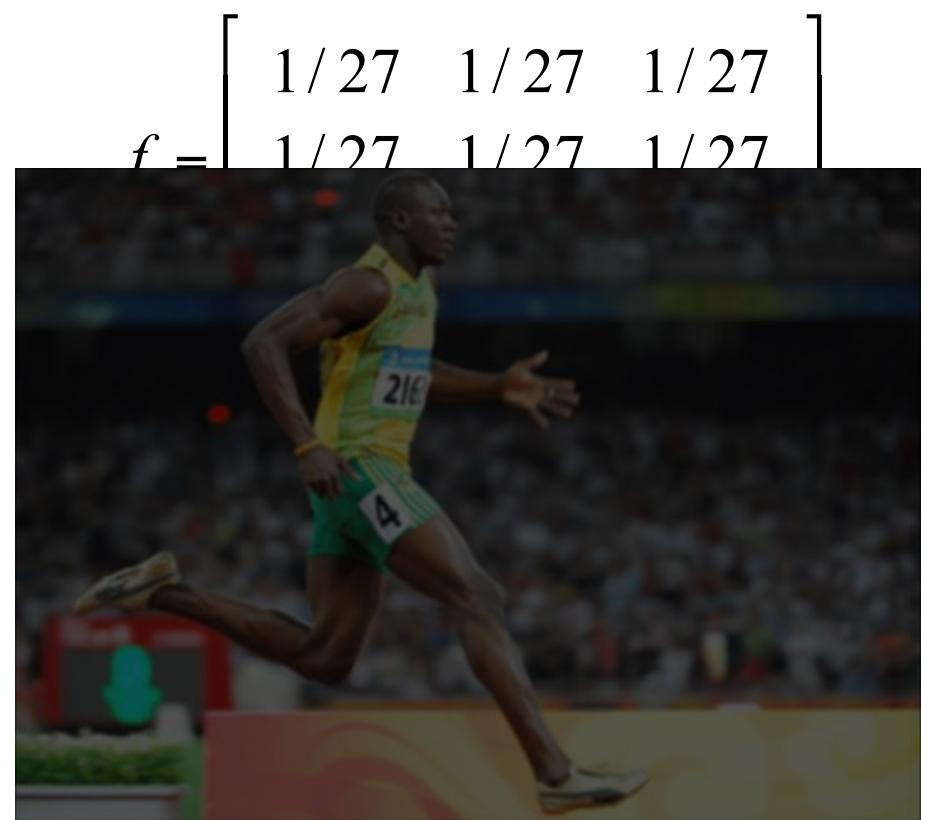
Imagine filtrată $f * I$

Filtrarea imaginilor color

PYTHON: $f = \text{np.ones}((3,3))/27$
 $F = \text{cv2.filter2D(img,-1,f)}$



Imagine inițială I



Imagine filtrată $f * I$

Aplicație: Imagini hibrid

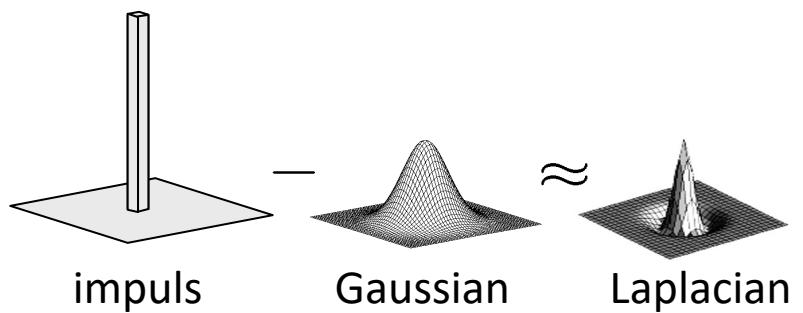
filtru Gaussian

(low pass filter – obține frecvențe joase)



filtru Laplacian

(high pass filter – obține frecvențe înalte)



A. Oliva, A. Torralba, P.G. Schyns,
“Hybrid Images” SIGGRAPH 2006

