

# Concepțe și aplicații în Vederea Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

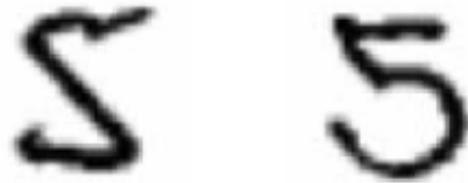
Radu Ionescu

[radu.ionescu@fmi.unibuc.ro](mailto:radu.ionescu@fmi.unibuc.ro)

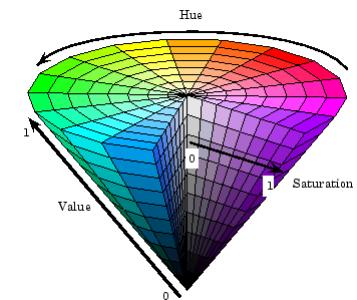
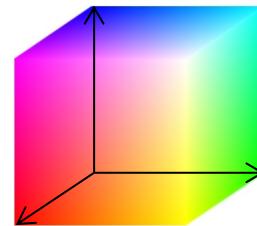
Curs optional  
anii III/IV, semestrul I, 2024-2025

# Cursul trecut

- Compararea contururilor

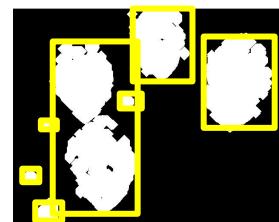
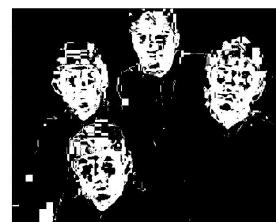


- Spațiile de culori RGB și HSV



- Imagini binare

- aplicarea unui prag
- operatori morfologici
- componente conexe
- proprietăți



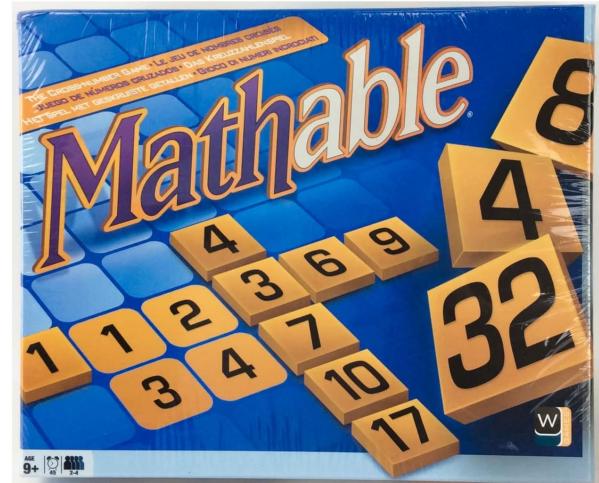
- Recunoașterea claselor de obiecte

- privire de ansamblu
- ce este o clasă de obiecte?

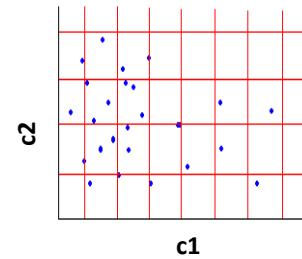


# Cursul de azi

- Tema 1: Calculator automat de scor pentru jocul Mathable



- Clasificarea imaginilor
  - reprezentarea caracteristicilor prin histograme
- Localizarea claselor de obiecte
  - metoda ferestrei glisante



# Tema 1 – Calculator automat de scor pentru jocul Mathable



<https://tinyurl.com/CAVA-2024-TEMA1>

# Mathable

Mathable este un joc bazat pe ecuații matematice care trebuie formate pe tabla de joc. A fost descris ca fiind similar cu jocul Scrabble, dar folosind numere (Figura 1). În Mathable, jucătorii folosesc o tablă de joc cu pătrate normale (pătrate albe), pătrate care impun anumite constrângeri (pătrate albastre) și pătrate care conțin un bonus (pătrate mov marcate  $2\times$  și pătrate portocalii marcate  $3\times$ ), precum și 106 piese cu numere. Jocul poate fi jucat de doi, trei sau patru jucători, scopul fiind formarea de ecuații matematice valide și acumularea unui punctaj cât mai mare.



Figura 1: Tabla de joc Mathable (stânga) și toate piesele posibile plasate pe tablă (dreapta).

# Tabla de joc

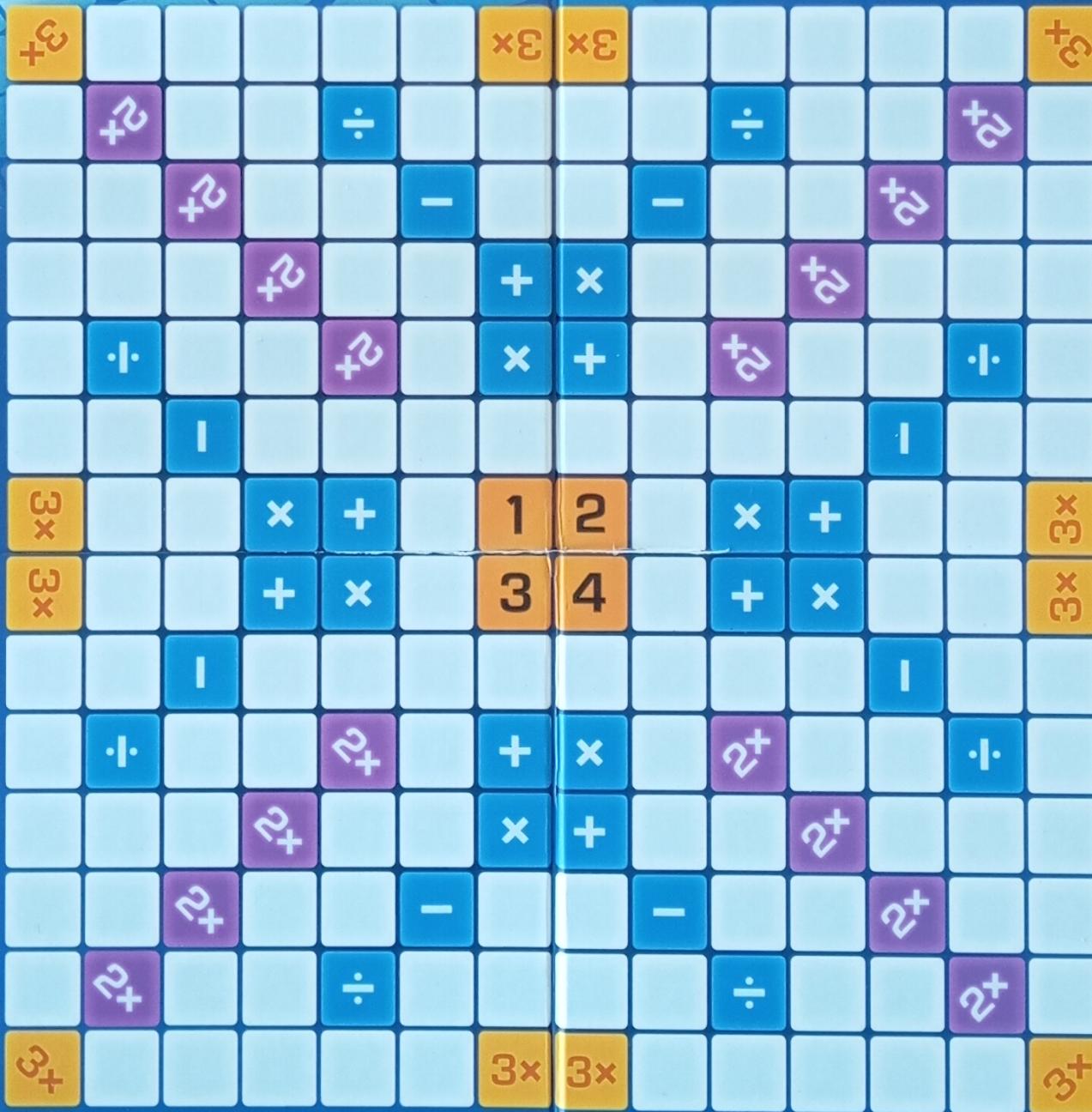
Tabla de joc Mathable (Figura 1) este împărțită într-o grilă de  $14 \times 14$  pătrate. Tabla este marcată cu pătrate colorate, unele dintre ele conținând un bonus, astfel încât cresc punctajul, iar altele impunând constrângeri. Există în total 28 pătrate colorate care oferă un bonus:

- 16 pătrate mov pentru puncte duble;
- 12 pătrate portocalii pentru puncte triple.

Pătratele cu constrângeri sunt marcate cu albastru și conțin un operator (binar): adunare, scădere, înmulțire sau împărțire. Pentru a ocupa un pătrat specific plasând o piesă acolo, jucătorul trebuie să formeze o ecuație care corespunde operatorului aceluia pătrat. În total, există 32 de pătrate cu constrângeri.



# Math



# Piese de joc

Pentru a forma ecuații matematice pe tablă, jucătorii folosesc piese de joc cu numere. Pentru simplitate, le vom numi *piese*. Există în total 46 piese diferite ce pot ajuta la formarea unor ecuații matematice. Piese sunt:

- cifrele 0-9 (pentru un total de 10 piese în intervalul [0,9]);
- numerele 10-19 (pentru un total de 10 piese în intervalul [10,19]);
- numerele 20, 21, 24, 25, 27, 28 (pentru un total de 6 piese în intervalul [20,29]);
- numerele 30, 32, 35, 36 (pentru un total de 4 piese în intervalul [30,39]);
- numerele 40, 42, 45, 48, 49 (pentru un total de 5 piese în intervalul [40,49]);
- numerele 50, 54, 56 (pentru un total de 3 piese în intervalul [50,59]);
- numerele 60, 63, 64 (pentru un total de 3 piese în intervalul [60,69]);
- numerele 70, 72 (pentru un total de 2 piese în intervalul [70,79]);
- numerele 80, 81 (pentru un total de 2 piese în intervalul [80,89]);
- numărul 90 (pentru un total de 1 piesă în intervalul [90,99]).

Numărul de piese disponibile în joc este specificat pe marginea din dreapta a tablei. Pentru fiecare valoare între 1 și 10 există 7 piese în joc (pentru un total de 70 de piese), în timp ce pentru restul de 36 de valori există câte o singură piesă cu acea valoare (pentru un total de 36 de piese). Așadar, în total, există 106 piese în joc.

# Piese de joc



# Piese de joc



# Configurații valide și invalide

O ecuație matematică este formată prin adunarea, scăderea, înmulțirea sau împărțirea a două piese adiacente, plasând o piesă cu rezultatul pe următorul patrat liber, fie în dreapta sau stânga, fie deasupra sau sub cele două piese originale, niciodată pe diagonală sau între piese. Figura 2 prezintă configurații valide și invalide pe o tablă generală Mathable. Primul jucător poate alege una dintre cele patru operații matematice folosind numerele 1 până la 4 inscripționate în centrul tablei de joc.

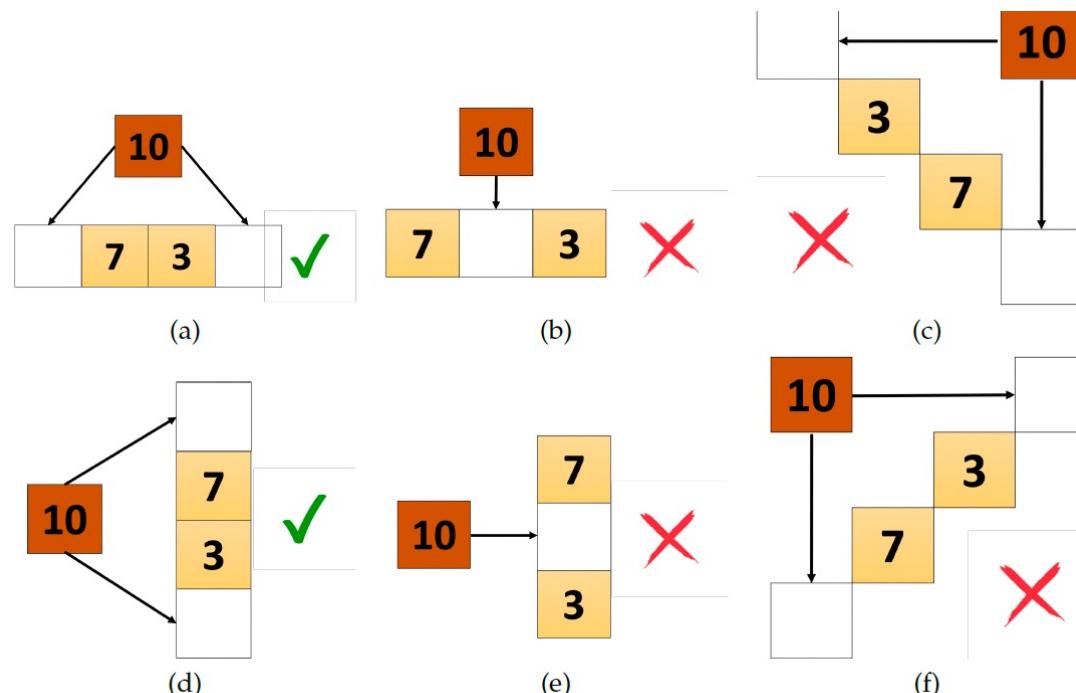


Figura 2: Configurații valide și invalide în Mathable. Exemplele (a) și (d) ilustrează mutări valide, unde piesa 10 poate fi plasată fie în dreapta sau în stânga (cazul (a)), fie deasupra sau dedesubtul celor două piese originale 7 și 3, completând adunarea. Exemplele (b) și (e) prezintă mutări invalide, unde piesa 10 nu poate fi plasată orizontal între piese (cazul (b)) sau vertical între piese (cazul (e)). Exemplele (c) și (f) prezintă mutări invalide, unde piesa 10 nu poate fi plasată pe diagonală față de piesele 7 și 3.

# Desfășurarea jocului

Considerăm scenariul cu doi jucători, Jucătorul 1 și Jucătorul 2. Fiecare jucător primește la începutul jocului 7 piese trase dintr-un sac cu toate piesele (înțial sunt 106 piese în sac). La fiecare rundă, jucătorul curent poate efectua mai multe mutări formând ecuații matematice pe tablă. După fiecare mutare, jucătorul curent obține un punctaj. Numărul maxim de mutări per rundă este limitat la 7 dar un jucător poate face mai puține mutări dacă nu mai poate forma ecuații matematice pe tabla de joc. La sfârșitul rundei în care a pus piese, jucătorul trage aleator piese din sac pentru a-și aduce numărul de piese în mână înapoi la 7. Scorul obținut de jucătorul curent în runda respectivă se obține prin adunarea scorurilor obținute la fiecare mutare.

*Primul jucător care mută poate alege una dintre cele patru operații matematice folosind numerele 1 până la 4 poziționate în centrul tablei de joc.*



# Calcularea punctajului

Scorul jucătorului curent după fiecare mutare este bazat pe piesa plasată pe tablă și poate crește în anumite cazuri.

**Valoarea punctajului unei piese.** Fiecare piesă are o valoare de punctaj egală cu numărul de pe piesă.

**Bonus pentru Ecuații Multiple.** Dacă o piesă, atunci când este plasată, completează mai multe ecuații, punctele sunt câștigate pentru fiecare ecuație completată.

**Pătrate cu premiu.** Un pătrat mov marcat 2× dublează numărul de puncte al piesei de pe acel pătrat, iar un pătrat portocaliu marcat 3× triplează numărul de puncte.

## Exemplu de scor

Figura 3 ilustrează douăsprezece mutări efectuate la începutul unui joc. Listăm mai jos cele douăsprezece mutări și oferim explicații detaliate despre calcularea scorului corespunzător fiecărui jucător după fiecare mutare. Cei doi jucători din acest exemplu au următoarele piese:

**Jucătorul 1:** 1, 2, 8, 12, 16, 17, 42;

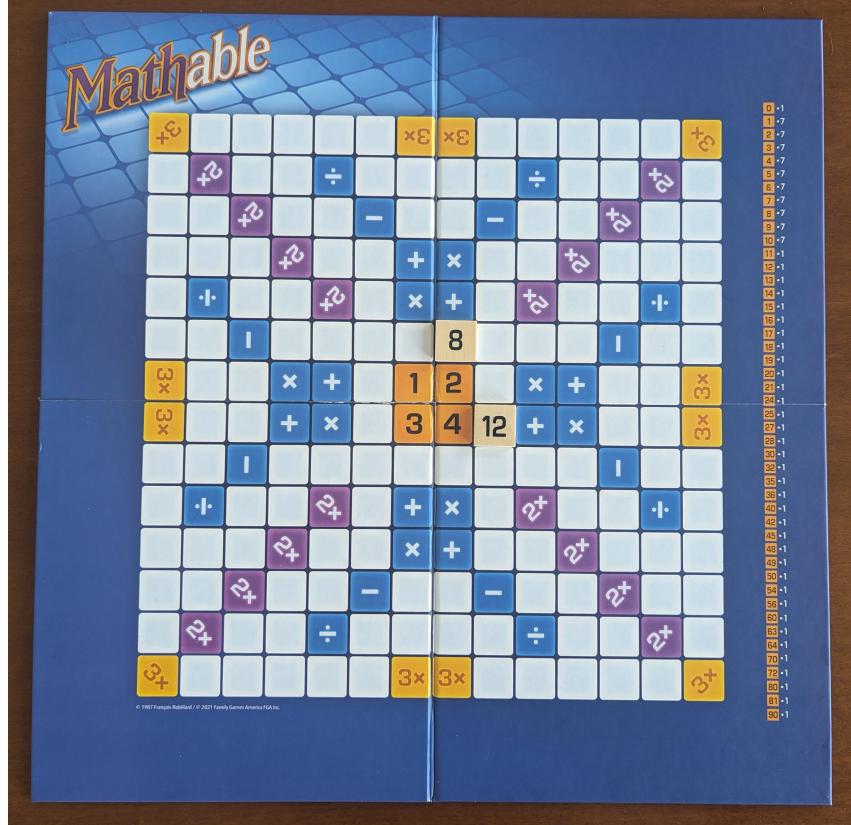
**Jucătorul 2:** 3, 4, 6, 7, 7, 8, 21.

# Example de mutări și calculare a scorului



**Mutarea 1.** Jucătorul 1 poate alege una dintre cele patru operații matematice folosind numerele 1 până la 4 poziționate în centrul tablei de joc. Jucătorul 1, care are o piesă 12 în mână, alege să înmulțească numerele 3 și 4. Poate plasa piesa 12 fie în stânga lui 3, fie în dreapta lui 4. Alege să o plaseze în dreapta. Întrucât ecuația este corectă, jucătorul câștigă 12 puncte.

# Example de mutări și calculare a scorului



**Mutarea 2.** Jucătorul 1 plasează a două piesă. Deoarece are un 8, decide să înmulțească 2 și 4 și alege să plaseze rezultatul deasupra lui 2. O mutare alternativă ar fi fost să poziționeze piesa 8 sub 4. Astfel, jucătorul 1 câștigă alte 8 puncte.

# Example de mutări și calculare a scorului



**Mutarea 3.** Jucătorul 1 plasează a treia piesă. Ia piesa 2 din mâna, scade 1 din 3 și plasează 2 sub 3. Ar putea, de asemenea, să plaseze 2 deasupra lui 1, dar alege să nu o facă. Jucătorul adaugă 2 puncte la totalul său.

# Example de mutări și calculare a scorului



**Mutarea 4.** Jucătorul 1 plasează a patra piesă. Scade 1 din 2 și plasează piesa 1 în stânga lui 1 deja aflat pe tablă. Ar putea, de asemenea, să o plaseze în partea dreaptă. Astfel, jucătorul 1 câștigă încă 1 punct.

# Example de mutări și calculare a scorului



**Mutarea 5.** Jucătorul 1 plasează a cincea piesă. În acest moment, pătratul din dreapta celui ocupat de piesa 12 este albastru, marcat cu operatorul +, ceea ce înseamnă că jucătorul trebuie să adune cele două piese pentru a ocupa acest pătrat. Jucătorul ia piesa 16 din mâna și o plasează pe pătratul albastru, îndeplinind astfel constrângerea. Jucătorul adaugă 16 puncte la scorul său. Jucătorul nu poate plasa altă piesă, astfel încât runda sa se încheie. Adună toate punctele câștigate în această rundă și obține un scor total de 39 de puncte.

# Example de mutări și calculare a scorului



**Mutarea 6.** Tura jucătorului 2. Jucătorul 2 decide să adune 3 și 4, deoarece are un 7 în mână. Plasează piesa în stânga lui 3 și primește 7 puncte.

# Example de mutări și calculare a scorului



**Mutarea 7.** Jucătorul 2 plasează a doua piesă. Scăzând 1 din 7 obține un total de 6. Plasează piesa 6 sub 7 (ar putea, de asemenea, să o plaseze deasupra lui 1). Adaugă 6 puncte la totalul său.

# Example de mutări și calculare a scorului



**Mutarea 8.** Jucătorul 2 plasează a treia piesă. Jucătorul decide să adune 6 și 2, obținând 8. Are două posibilități: plasarea lui 8 în stânga lui 6, câștigând 8 puncte sau plasarea lui 8 în dreapta lui 2, profitând astfel de Bonusul pentru Ecuații Multiple, deoarece vertical,  $2 \times 4$  fac de asemenea 8. Jucătorul optează pentru a doua alegere, adăugând încă 8 puncte (un total de 16) la scorul său. REMARCĂ: Dacă cele două piese deasupra acestui pătrat nu completează ecuația, jucătorul poate totuși să plaseze o piesă acolo, contabilizând doar 8 puncte pentru adunare.

# Example de mutări și calculare a scorului



**Mutarea 9.** Jucătorul 2 plasează a patra piesă. De această dată folosește piesa 4, adunând 3 și 1. Plasează piesa deasupra lui 1. Câștigă 4 puncte.

# Example de mutări și calculare a scorului



**Mutarea 10.** Jucătorul 2 plasează a cincea piesă. Pentru a folosi piesa 3, împarte 6 la 2. Rezultatul 3 este plasat lângă 6. Adaugă 3 puncte.

# Example de mutări și calculare a scorului



**Mutarea 11.** Jucătorul 2 plasează a șasea piesă. Înmulțind 7 cu 3 obține 21. Această piesă este plasată în stânga lui 7. Adaugă 21 de puncte. De asemenea, rețineți că, din nou, jucătorul curent îndeplinește constrângerea pătratului albastru pe care a plasat piesa.

# Example de mutări și calculare a scorului



**Mutarea 12.** Jucătorul 2 își plasează a șaptea piesă. Plasează piesă 7 sub piesă 3, deoarece  $21 \div 3 = 7$ . Nu poate plasa piesă deasupra piesei 21, deoarece un pătrat de constrângere nu îi permite să facă împărțirea. Plasând piesă 7 sub piesă 3, ajunge pe un pătrat cu premiu ( $2\times$ ), care dublează punctele câștigate în această mutare la  $14 = 7 \times 2$ . Punctajul total obținut în runda sa este astfel 71.

# Descrierea datelor

Arhiva cu materiale (disponibilă aici <https://tinyurl.com/CAVA-2024-TEMA1>) conține patru directoare: *antrenare*, *testare*, *evaluare* și *imagini\_auxiliare*. Directoarele *antrenare* și *testare* au aceeași structură, deși datele din directorul *testare* vor fi disponibile după termenul limită de trimitere a codului soluției (**prima fază** - detalii mai jos).

Datele constau în imagini cu tabla de joc Mathable și cu piesele de Mathable puse pe tablă după fiecare mutare. Toate imaginile sunt realizate cu același telefon mobil aflat într-o poziție fixă deasupra tablei de joc. Este posibil ca luminozitatea scenei (a tablei de joc) să prezinte mici variații de la o imagine la alta.

Directorul *imagini\_auxiliare* conține câteva imagini cu tabla de joc cu: (i) toate piesele aranjate în diverse configurații; (ii) fără piese. Folosiți aceste imagini pentru a înțelege mai bine problema și pentru a extrage date pentru soluția voastră.

Directorul *antrenare* conține datele de antrenare constând în imagini de la primele 50 de mutări pentru 4 jocuri de Mathable. În total sunt 200 de imagini de antrenare și 200 de fișiere cu adnotări corespunzătoare fiecărei imagini. Imaginea de antrenare  $i$  din jocul  $j$  este denumită ' $j\_i.jpg$ ', unde  $j \in \{1, 2, 3, 4\}$  și  $i \in \{01, 02, 03, \dots, 50\}$ . Fișierul cu adnotări corespunzător are același format, extensia ' $.jpg$ ' fiind înlocuită de extensia ' $.txt$ '. Pentru fiecare joc, există un fișier  $.txt$  care indică începutul fiecărei ture pentru un jucător, denumit  $g\_turns.txt$ , unde  $g$  reprezintă numărul jocului. Fișierul  $.txt$   $g\_scores.txt$ , corespunzător aceluiași joc, conține scorurile actualizate după fiecare rundă, reflectând performanța jucătorilor pe parcursul jocului, extinzând informațiile din fișierul  $g\_turns.txt$ .

# Descrierea datelor

 CAVA-2024-Tema1	 >	
 antrenare	 >	
 evaluare	 >	
 imagini_auxiliare	 >	
 README.txt	 >	
 Tema1.pdf	 >	
 testare	 >	

# Descrierea datelor

CAVA-2024-Tema1	✓ >		
		antrenare	✓ >
		evaluare	✓ >
		imagini_auxiliare	✓ >
		README.txt	✓
		Tema1.pdf	✓
		testare	✓ >
		1_01.jpg	✓
		1_01.txt	✓
		1_02.jpg	✓
		1_02.txt	✓
		1_03.jpg	✓
		1_03.txt	✓
		1_04.jpg	✓
		1_04.txt	✓
		1_05.jpg	✓
		1_05.txt	✓
		1_06.jpg	✓
		1_06.txt	✓
		1_07.jpg	✓
		1_07.txt	✓
		1_08.jpg	✓
		1_08.txt	✓
		1_09.jpg	✓

# Descrierea datelor

 CAVA-2024-Tema1  >	 antrenare  >  <b>evaluare</b>  >  imagini_auxiliare  >  README.txt   Tema1.pdf   testare  >	 cod_evaluare  >  fake_test  >  fisiere_solutie  >
---	---	--

# Descrierea datelor

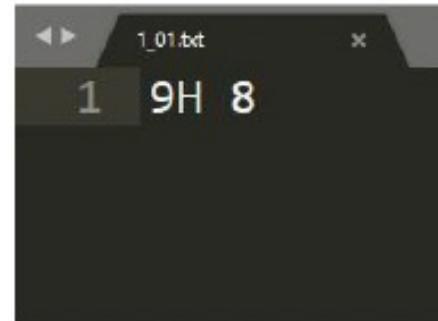
 CAVA-2024-Tema1 <span style="float: right;">✓ &gt;</span>	 antrenare <span style="float: right;">✓ &gt;</span>	 01.jpg <span style="float: right;">✓</span>
	 evaluare <span style="float: right;">✓ &gt;</span>	 02.jpg <span style="float: right;">✓</span>
	 <b>imagini_auxiliare</b> <span style="float: right;">✓ &gt;</span>	 03.jpg <span style="float: right;">✓</span>
	 README.txt <span style="float: right;">✓</span>	 04.jpg <span style="float: right;">✓</span>
	 Tema1.pdf <span style="float: right;">✓</span>	
	 testare <span style="float: right;">✓ &gt;</span>	

# Adnotări

Fișierele cu adnotări conțin următoarele informații:

- poziția piesei adăugate pe tablă, specificată în ordine de la stânga la dreapta și de sus în jos. Poziția este indicată folosind numerele 1-14 pentru rânduri și literele A-N pentru coloane. Trebuie menționat că doar o piesă este plasată la fiecare mutare.
- numărul de pe piesa corespunzător poziției respective de pe tablă.

În Figura 4 puteți vedea un exemplu de fișier de adnotare pentru imaginea 1\_01.jpg.



# Evaluare

Directorul *evaluare* vă indică cum să vă scrieți codul astfel încât să respectați formatul impus pentru fișierele cu rezultate folosite în faza de evaluare pe datele de test (**faza a doua**) ce va avea loc după trimiterea codului cu soluția de fiecare din voi. Conține următoarele sub-directoare:

- *fake\_test* - acest director exemplifică cum vor arăta datele de testare (pentru un singur joc), el păstrează aceeași structură ca cea descrisă pentru directorul *antrenare* descris anterior. Acest director va fi similar cu directorul *testare* în care vom pune imaginile de testare pentru faza a doua de evaluare (pentru patru jocuri).
- *fisiere\_solutie* - acest director exemplifică formatul fișierelor cu rezultatele pe care trebuie să le trimiteți în faza a doua. Veți trimite rezultatele voastre în acest format, încărcând o arhivă zip a unui director similar cu cel numit *331\_Alexe\_Bogdan*;
- *cod\_evaluare* - acest director conține codul care va fi folosit pentru evaluarea automată a rezultatelor voastre folosind adnotările soluțiilor corecte (ground-truth). Asigurați-vă că acest cod rulează pe fișierele voastre. Adnotările soluțiilor corecte vor fi disponibile după faza a doua.

# Evaluare

 antrenare	✓ >
 evaluare	✓ >
 imagini_auxiliare	✓ >
 README.txt	✓
 Tema1.pdf	✓
 testare	✓ >
<hr/>	
 cod_evaluare	✓ >
 fake_test	✓ >
 fisiere_solutie	✓ >
<hr/>	
 evaluateaza_solutie.py	✓
 fake_test_gt	✓ >

# Evaluare

antrenare	✓ >	cod_evaluare	✓ >	1_01.jpg	✓
evaluare	✓ >	fake_test	✓ >	1_02.jpg	✓
imagini_auxiliare	✓ >	fisiere_solutie	✓ >	1_03.jpg	✓
README.txt	✓			1_04.jpg	✓
Tema1.pdf	✓			1_05.jpg	✓
testare	✓ >			1_06.jpg	✓
				1_07.jpg	✓
				1_08.jpg	✓
				1_09.jpg	✓
				1_10.jpg	✓
				1_11.jpg	✓
				1_12.jpg	✓
				1_13.jpg	✓
				1_14.jpg	✓
				1_15.jpg	✓
				1_16.jpg	✓
				1_17.jpg	✓

# Evaluare

 cod_evaluare	 >	 331_Alexe_Bogdan	 >	 1_01.txt	
 fake_test	 >			 1_02.txt	
 fisiere_solutie	 >			 1_03.txt	
				 1_04.txt	
				 1_05.txt	
				 1_06.txt	
				 1_07.txt	
				 1_08.txt	
				 1_09.txt	
				 1_10.txt	
				 1_11.txt	
				 1_12.txt	
				 1_13.txt	
				 1_14.txt	
				 1_15.txt	
				 1_16.txt	
				 1_17.txt	

# Cerințe și notare

Scrieți un program în Python/Jupyter notebook care extrage automat informațiile vizuale după fiecare mutare dintr-un joc de Mathable și calculează scorul pentru runda unui jucător. Distribuția datelor de testare este aceeași ca la antrenare, imaginile fiind luate în aceleași condiții.

Pentru această temă, vom folosi următoarele reguli de notare:

- **Task 1 - 5 puncte** - vom evalua performanța algoritmului vostru pe o mulțime de 200 de imagini de testare ce reprezintă 4 jocuri de Mathable, fiecare joc având 50 de runde. Pentru fiecare imagine de testare algoritmul vostru trebuie să furnizeze poziția piesei care a fost plasată pe tablă în runda respectivă. Veți primi punctajul numai dacă algoritmul vostru furnizează corect poziția piesei adăugate pe tablă. Nu există punctaje parțiale. Fiecare configurație corectă valorează 0.025 puncte pentru un total de **5 puncte**;
- **Task 2 - 2 puncte** - vom evalua performanța algoritmului vostru pe aceleași 200 de imagini de la Task-ul 1, pentru piesa plasată trebuie să recunoașteți numărul care se află pe piesa respectivă. Veți primi punctajul numai dacă algoritmul vostru recunoaște corect numărul de pe piesă. Nu există punctaje parțiale. Pentru fiecare piesă pentru care recunoașteți corect numărul primiți 0.01 puncte pentru un total de **2 puncte**;
- **Task 3 - 2 puncte** - vom evalua performanța algoritmului vostru pe aceleași 200 de imagini de la Task-urile 1 și 2, pentru fiecare rundă a unui jucător trebuie să calculați scorul obținut de jucătorul curent. Scorul fiecărei runde se obține adunând scorurile după fiecare mutare din runda respectivă. Veți primi punctajul numai dacă algoritmul vostru calculează corect scorul pentru runda curentă. În fișierele adnotate pe

# Cerințe și notare

multimile de antrenare (*g\_turns.txt* și *g\_scores.txt*) și testare (*g\_turns.txt*) vor exista 50 de runde, aşa încât va trebui să calculați 50 de scoruri. Pentru fiecare scor corect calculat primiți 0.04 puncte pentru un total de **2 puncte**;

- **documentație - 1 punct** - descrieți într-un fișier pdf de minim două pagini soluția voastră pentru rezolvarea celor trei task-uri. Puteți ilustra aspecte cheie ale soluției voastre adăugând secvențe de cod și vizualizări ale imaginilor pentru soluția voastră. Acest fișier ar trebui să conțină suficientă informație astfel încât un student de nivel mediu de la cursul nostru să poată reimplementa soluția descrisă de voi.
- **oficiu - 1 punct** - primiți acest punct dacă formatul fișierelor voastre urmează formatul impus iar codul nostru de evaluare rulează pe datele primite de la voi fără a face modificări în fișierele voastre.

# Termen limită

## Termene limită

**Prima fază - trimiterea codului.** Încărcați o arhiva zip cu codul soluției voastre și un fișier pdf ce descrie soluția voastră până **marti, 3 decembrie, ora 23:59** la link-ul acesta <https://tinyurl.com/CAVA-2024-TEMA1-SOLUTII>. Includeți în arhiva zip NU-MAI cod (fișiere .py sau .ipynb) sau alte fișiere necesare rulării codului (fișiere cu modelele voastre antrenate, alte imagini auxiliare folosite de voi etc.). Nu includeți în arhiva voastră imaginile inițiale (le avem și noi!!!). **ATENȚIE: nu vom accepta teme trimise după data limită.**

Codul vostru ar trebui să includă un fișier README (vedeți exemplul din materiale) cu următoarele informații: (i) librăriile folosite de voi și necesare pentru rularea soluției voastre; (ii) indicații despre cum ar trebui rulat codul pentru fiecare task. Studenții care nu încarcă un fișier pdf cu descrierea soluției lor vor primi 0 puncte la partea de documentație.

**A doua fază - trimiterea rezultatelor.** Miercuri, 4 decembrie, vom publica datele de test în directorul *testare* de la adresa <https://tinyurl.com/CAVA-2024-TEMA1>. Veți rula soluția voastră pe imaginile de test și veți încărca rezultatele în aceeași zi ca o arhivă zip folosind următorul link <https://tinyurl.com/CAVA-2024-TEMA1-REZULTATE>.

# Recunoașterea claselor de obiecte

# Clasificarea imaginilor

- Conține o imagine test instantieri ale unei clase de obiecte/categorii ?
  - răspuns binar: DA/NU
- Exemplu: clasificator de imagini pentru ‘câine’

DA

NU

DA



# Clasificarea imaginilor

Învățăm un clasificator al imaginilor pe bază de exemple

**Exemple pozitive: imagini care conțin câini**

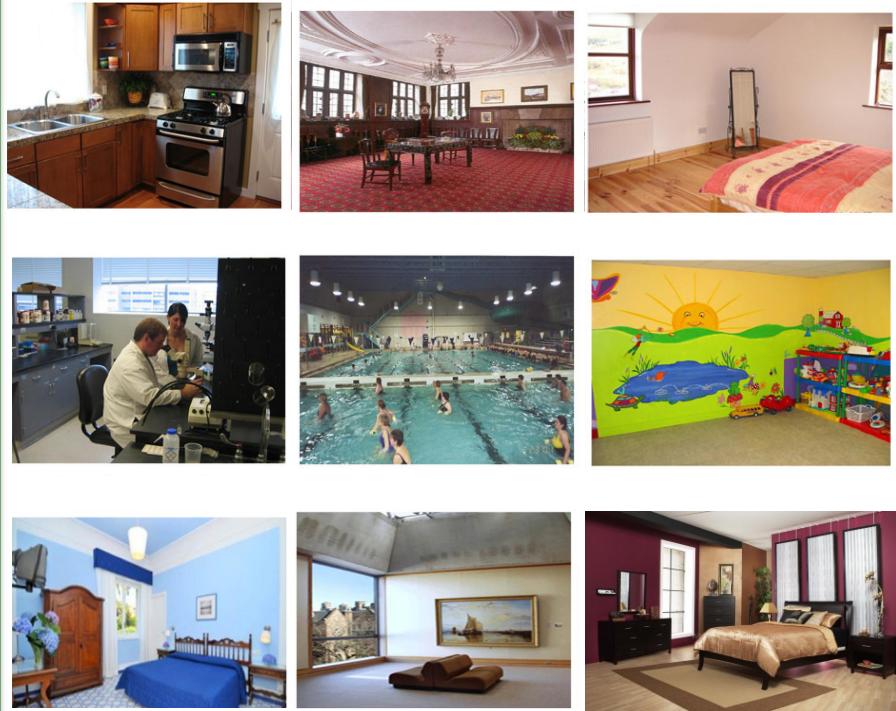


**Exemple negative: imagini care NU conțin câini**

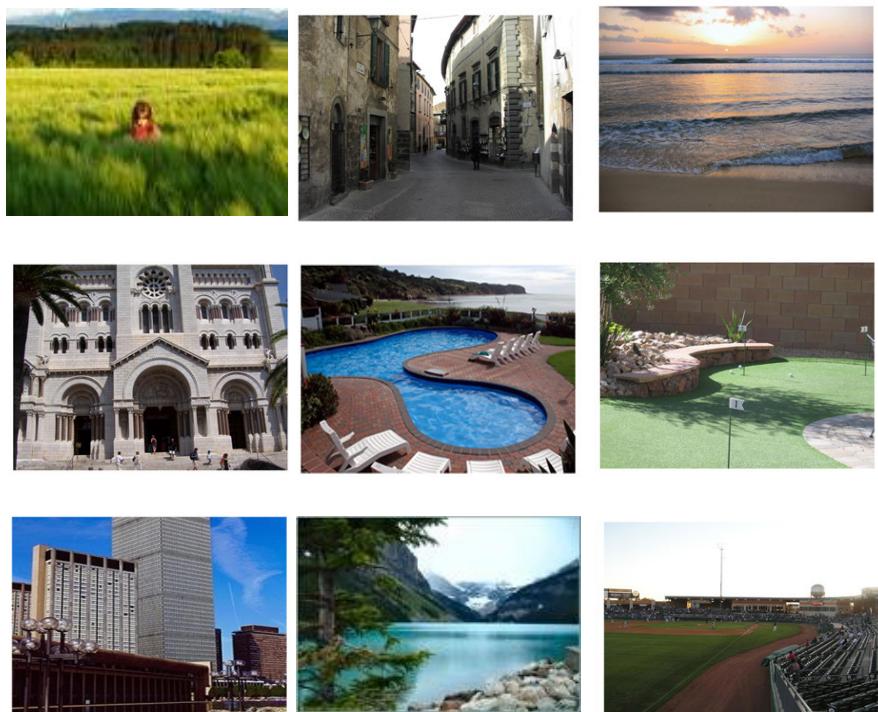


# Alt exemplu: 2 clase - indoor/outdoor

## Indoor – exemple



## Outdoor - exemple



Clasificare = asocierea unei etichete (indoor, outdoor) pentru exemple noi

outdoor



outdoor

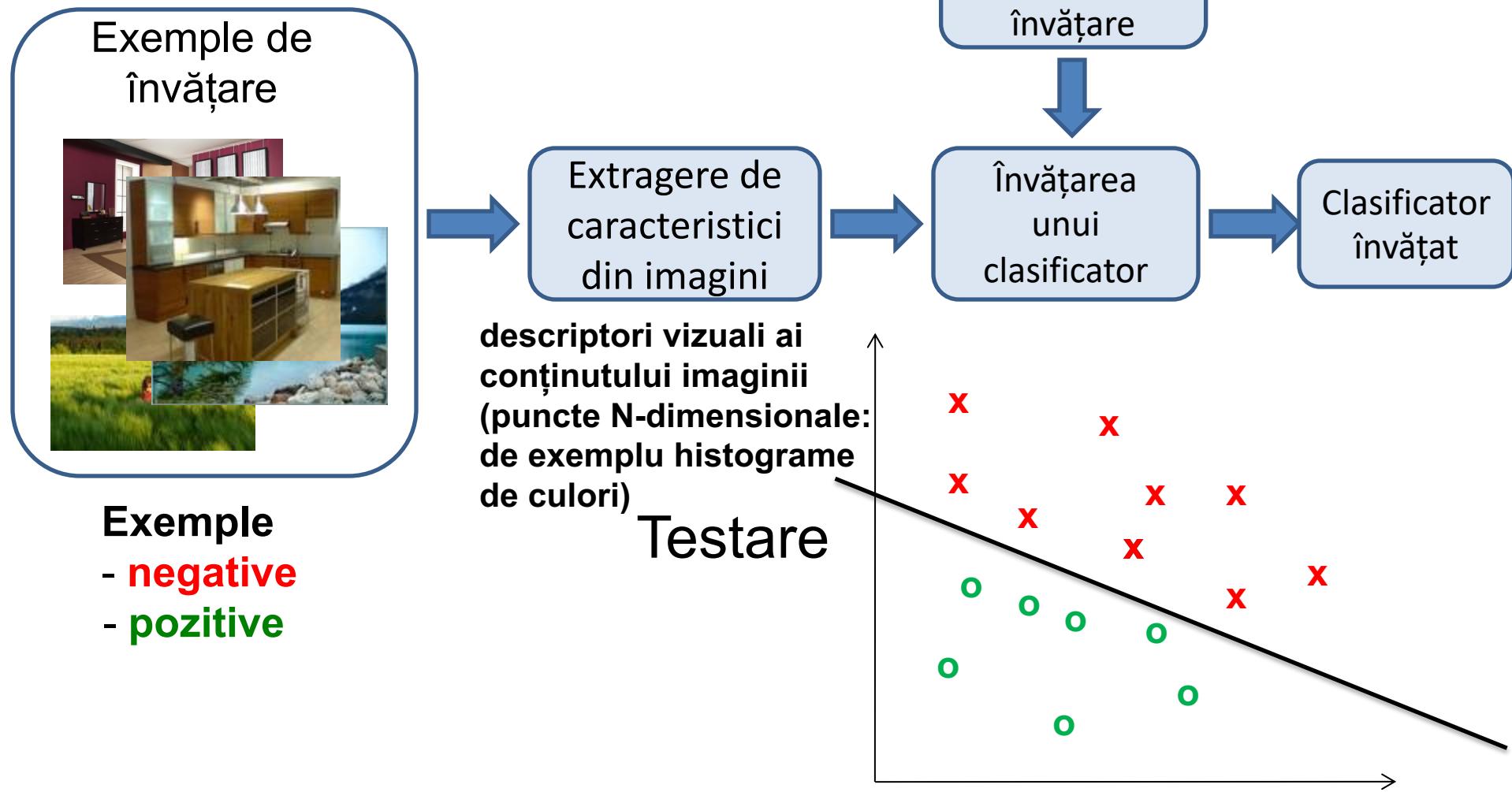


indoor

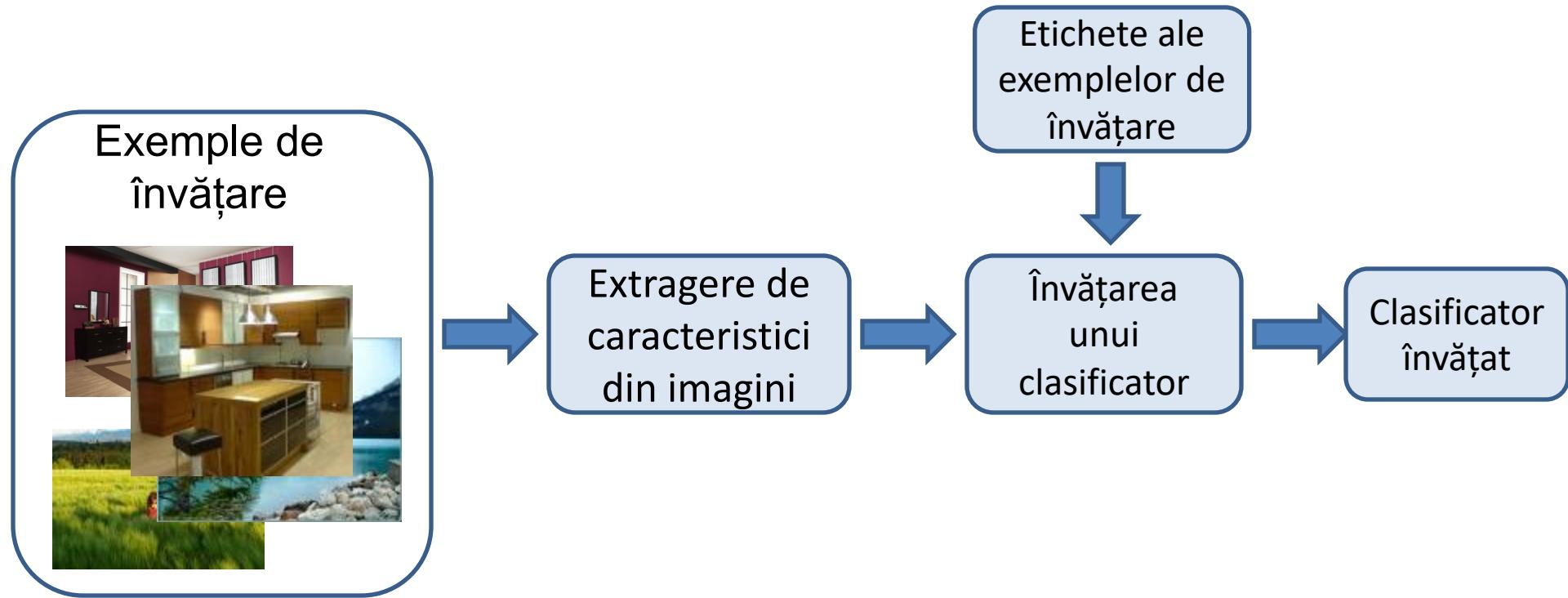


Imagini test:

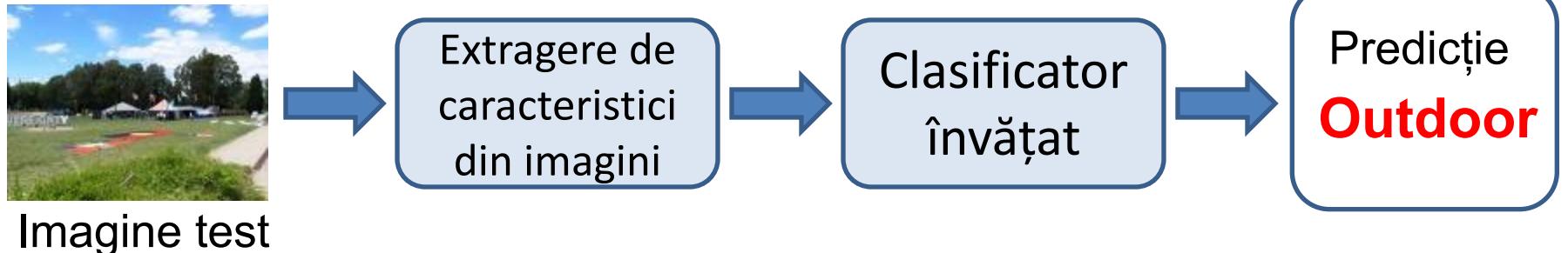
# Învățarea supervizată



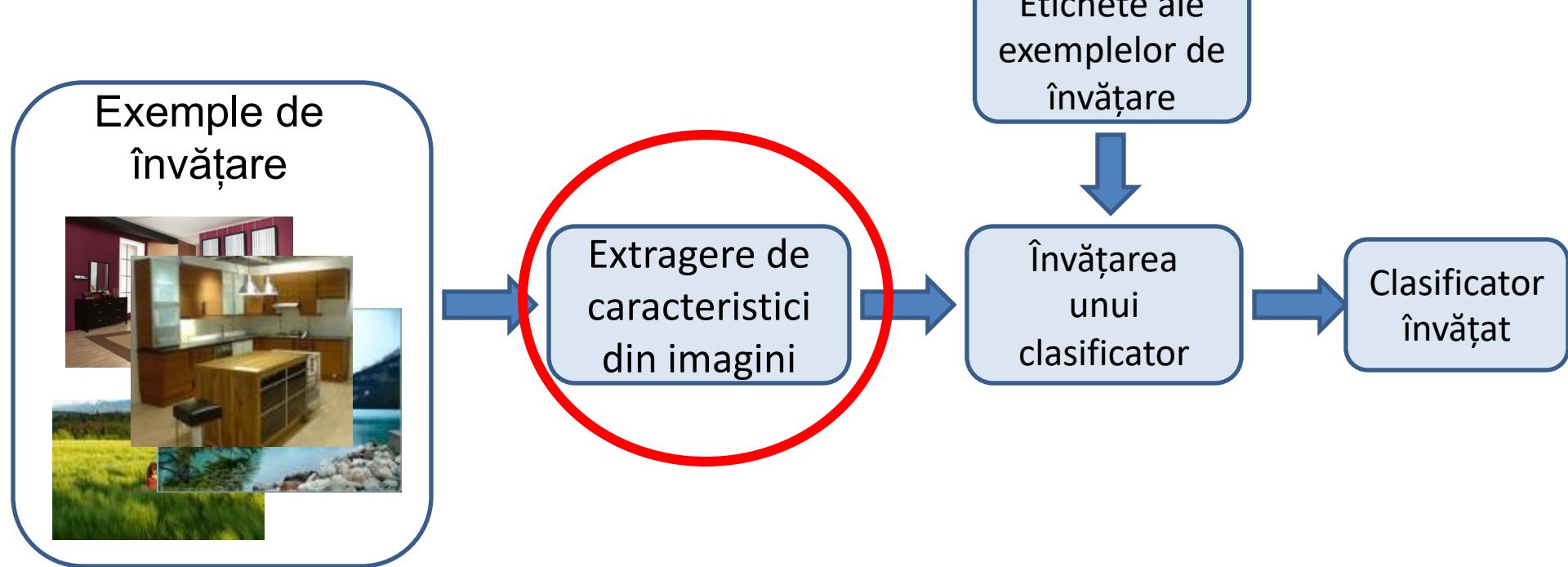
# Etapa de testare



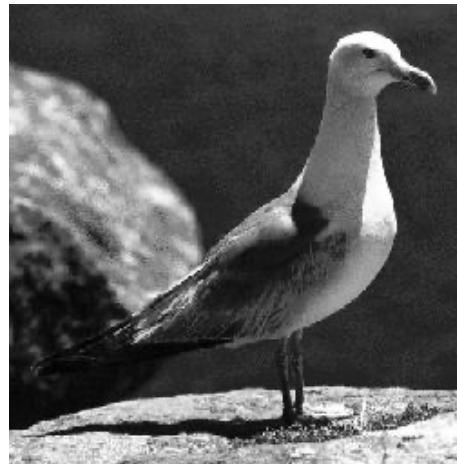
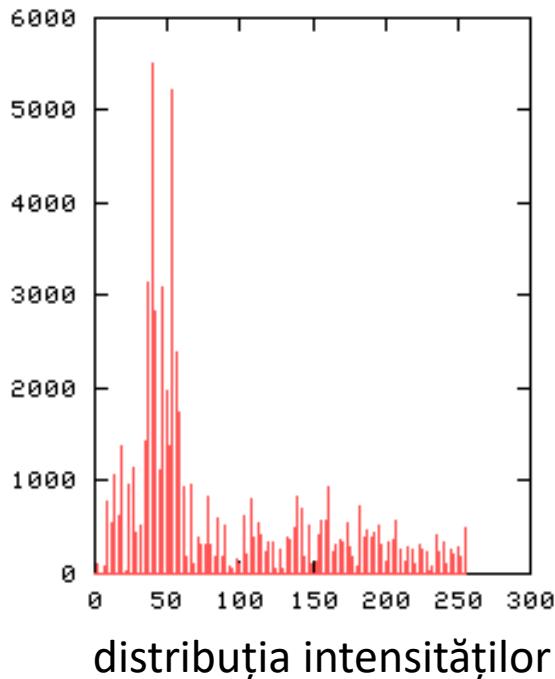
## Testare



# Extragere de caracteristici din imagini



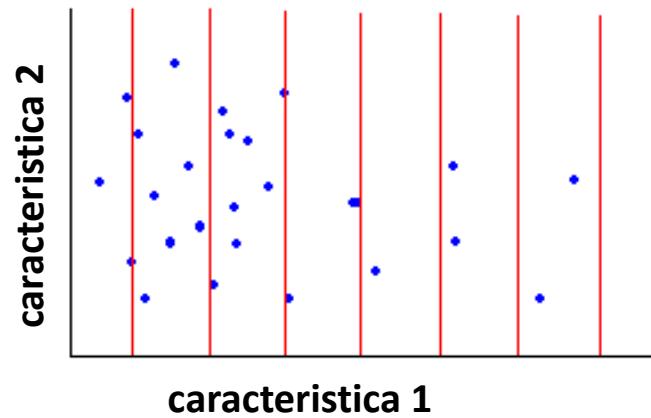
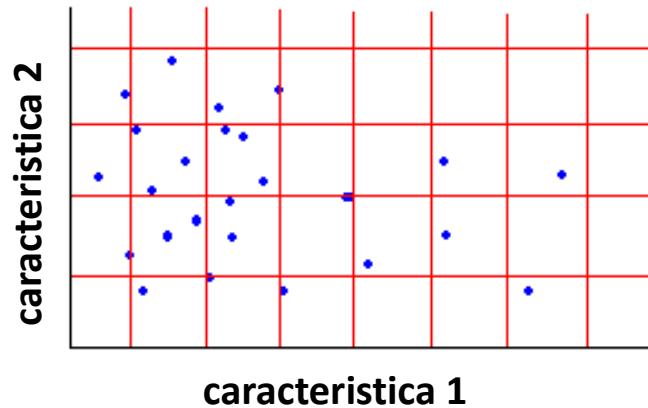
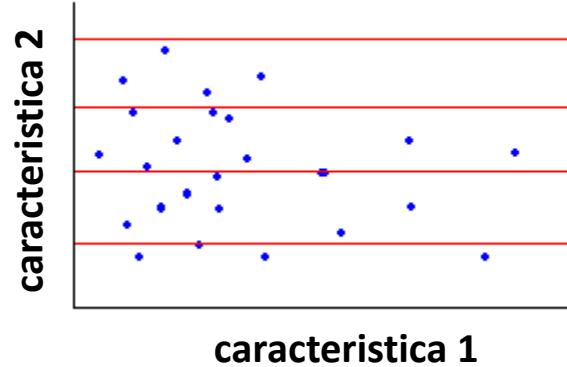
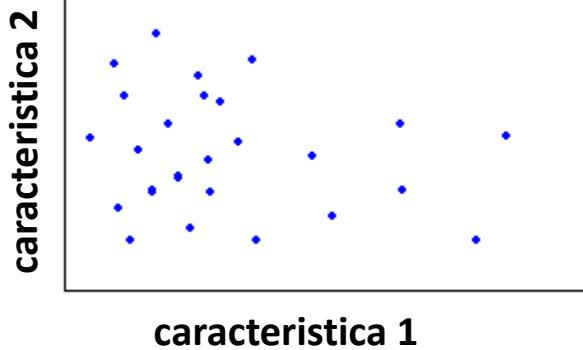
# Histograme pentru o caracteristică



Măsoară distribuția caracteristicilor (intensitate, culoare , textură):

- grupează datele în intervale și numără câte puncte “pică” în fiecare interval
- normalizează histograma ( suma elementelor = 1)

# Histograme pentru mai multe caracteristici



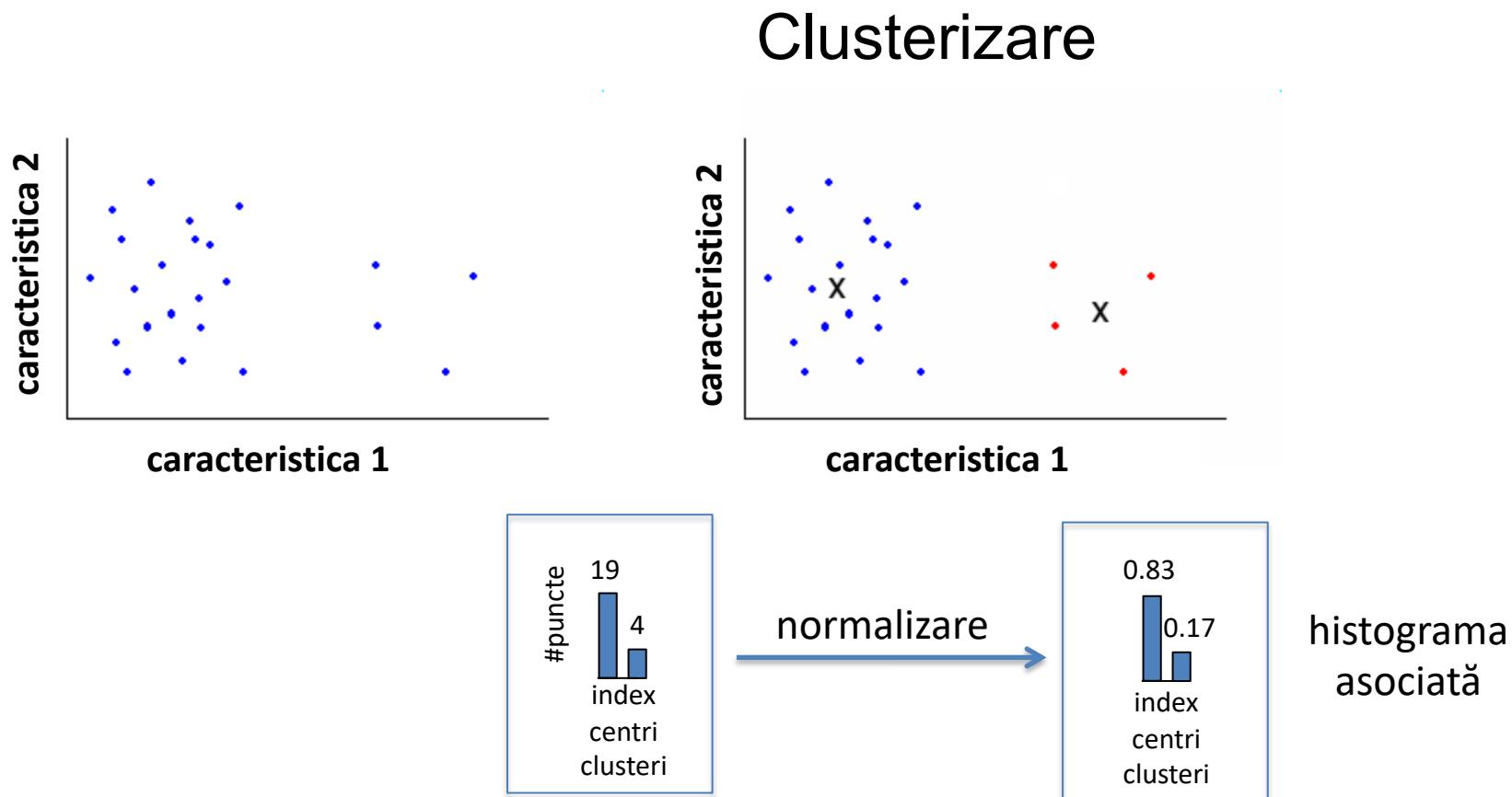
## Histograma comună

- necesită multe date pentru o aproximare bună a distribuției
- $\#intervale1 * \#intervale2$

## Histograma individuală

- necesită caracteristici independente
- mai multe puncte/interval decât în cazul histogramei comune

# Histograme pe baza clusterizării



Foloseşte aceeaşi centri ai clusterilor pentru toate imaginile

# Calculul distanței între histograme

- Distanța Euclidină (L2)

$$d^2(h_i, h_j) = \sum_{m=1}^K (h_i(m) - h_j(m))^2$$

- Distanța Chi-pătrat

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{m=1}^K \frac{(h_i(m) - h_j(m))^2}{h_i(m) + h_j(m)}$$

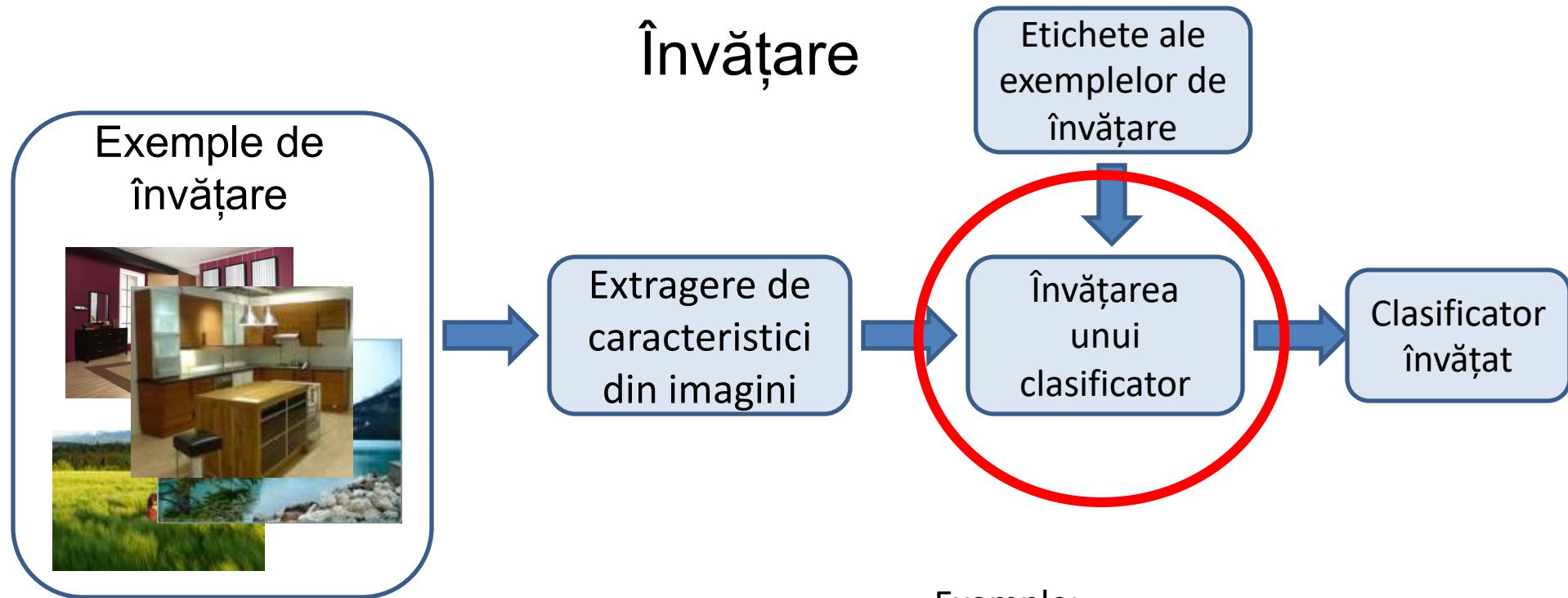
- Distanța dată de intersecția histogramelor

$$histint(h_i, h_j) = 1 - \sum_{m=1}^K \min(h_i(m), h_j(m))$$



Histograme de culori cele mai apropiate (distanța Chi-pătrat)

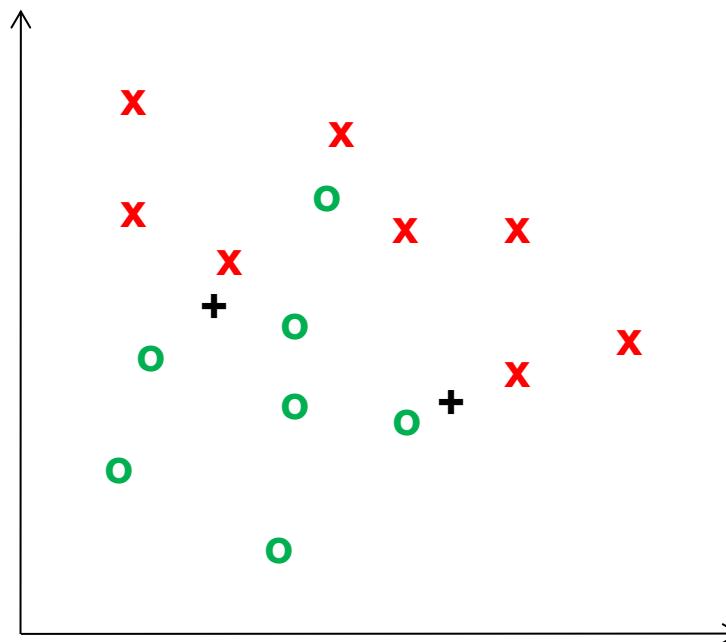
# Învățarea unui clasificator



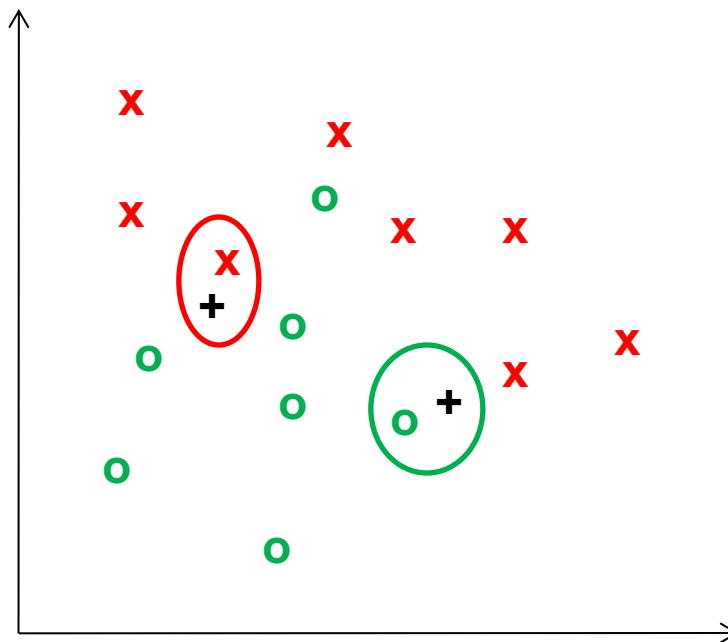
Exemplu:

- cei mai apropiati K-vecini
- clasificator liniar

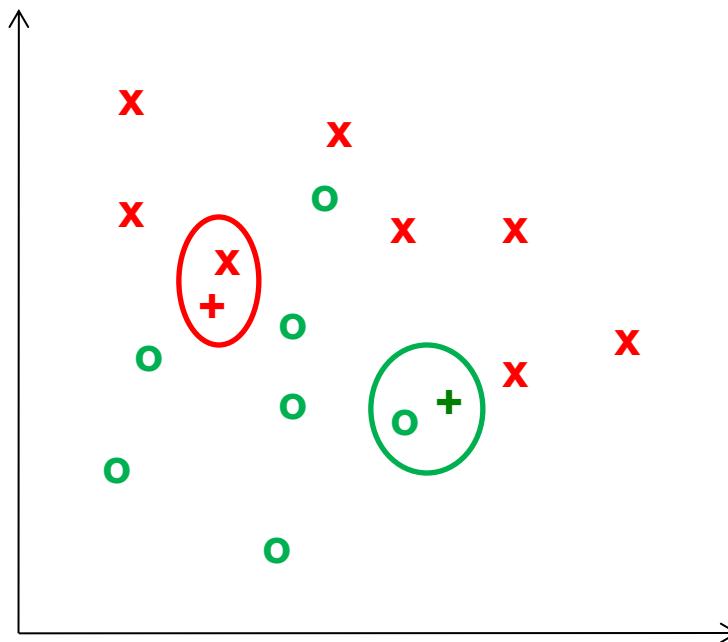
# Clasificatorul bazat pe cei apropiati K -vecini



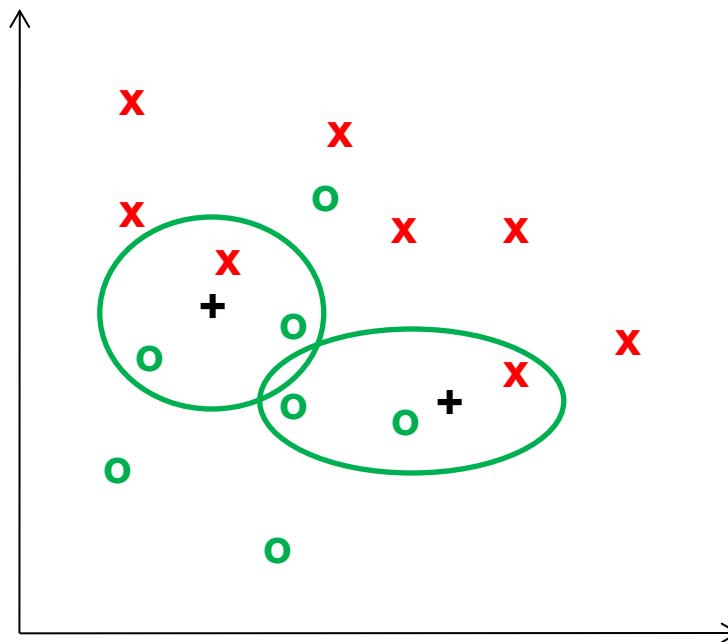
# K=1: cel mai apropiat vecini



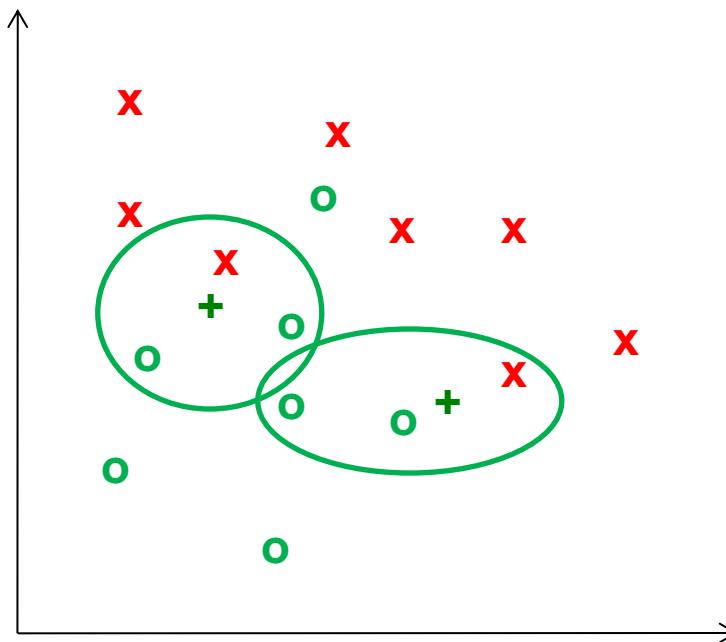
# $K=1$ : cel mai apropiat vecini



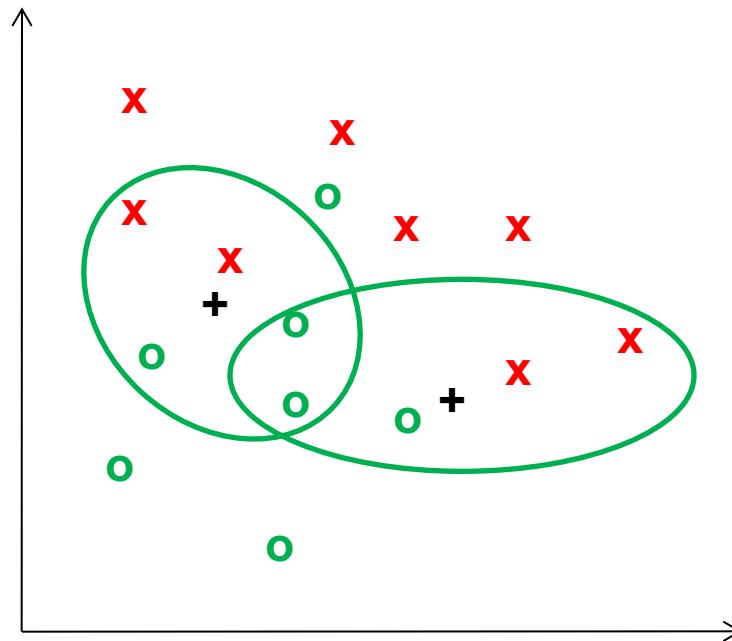
# K=3: cei mai apropiati 3 vecini



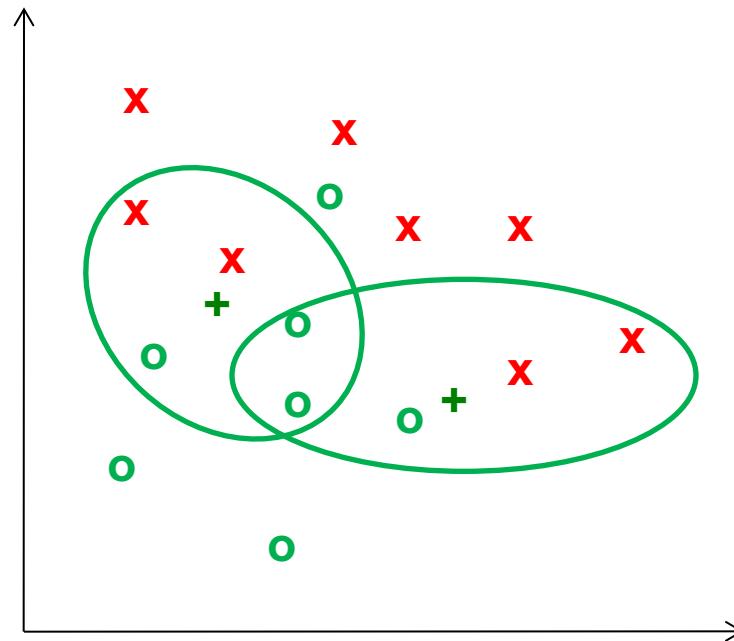
# K=3: cei mai apropiati 3 vecini



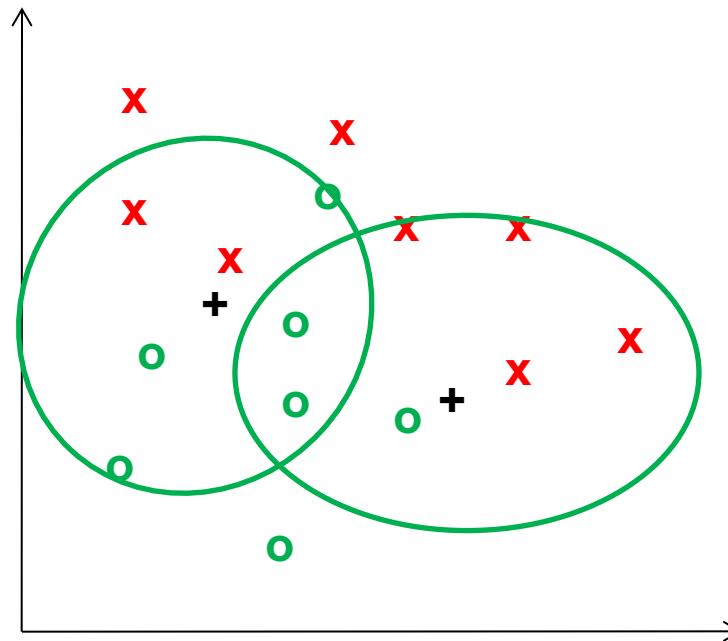
# K=5: cei mai apropiati 5 vecini



# K=5: cei mai apropiati 5 vecini

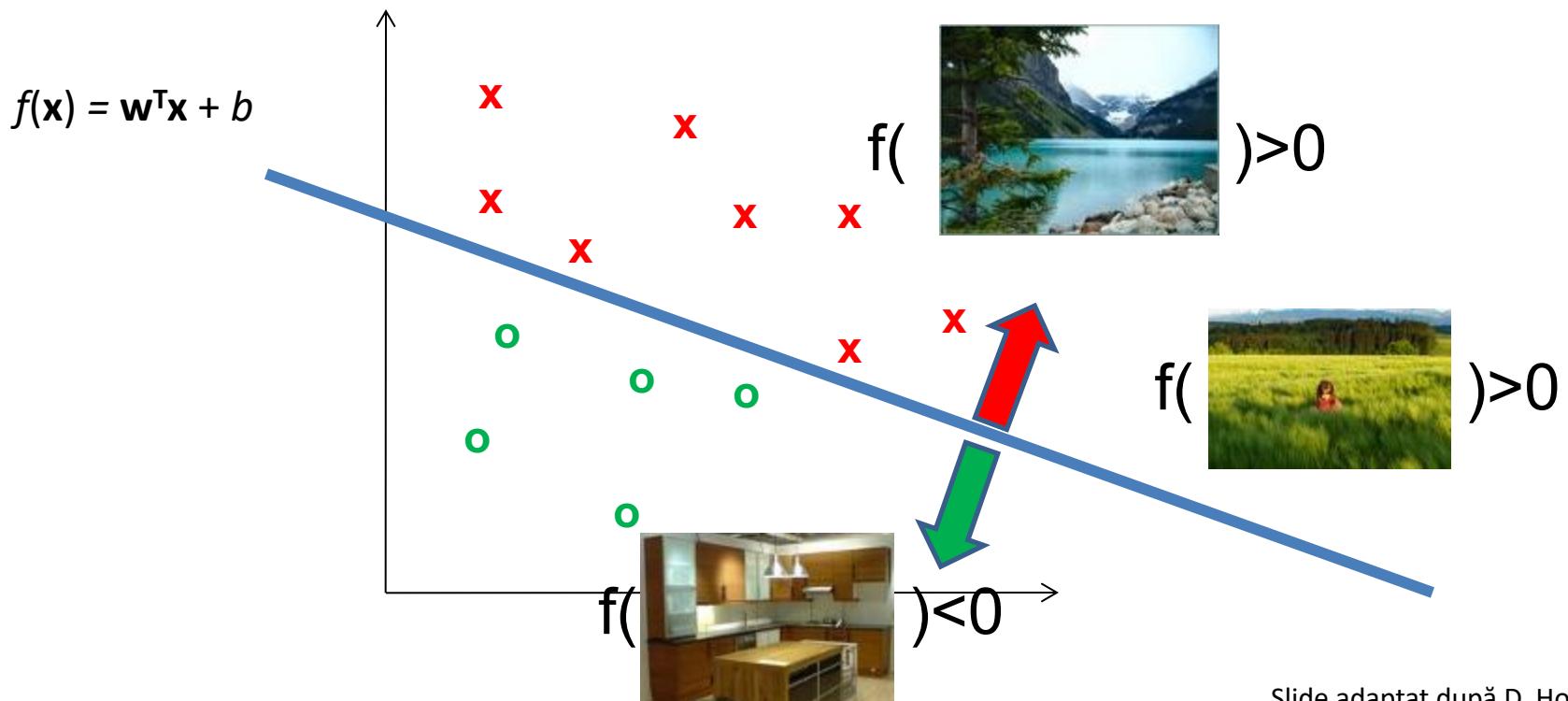


# Cei mai apropiati K-vecini



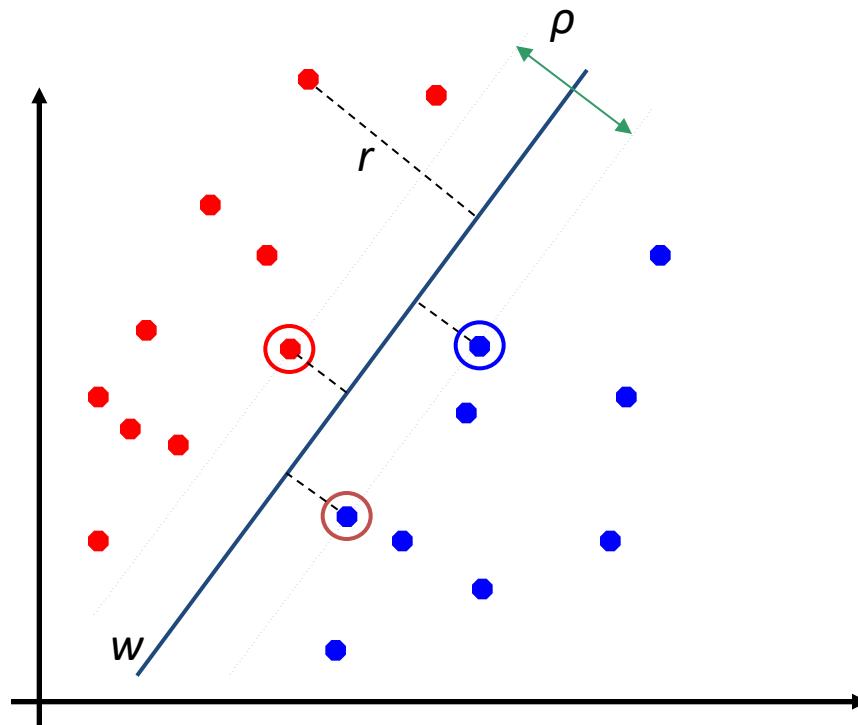
# Clasificatorul liniar

Găsește hiperplanul care separă exemplele din clase diferite



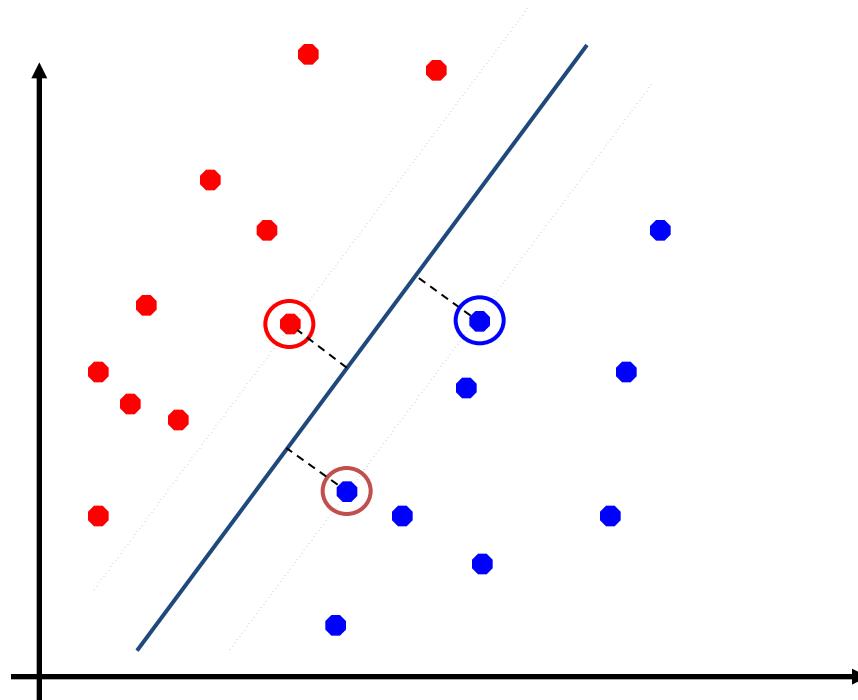
# Marginea de clasificare

- Distanța dintre exemplul  $\mathbf{x}_i$  și hiperplanul de separare este  $r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$
- Examplele cele mai aproape de hiperplan se numesc **vectori suport**.
- **Marginea**  $\rho$  dintre cele două clase = distanța dintre vectorii suport.



# Clasificatorul de margine maximă

- Numai vectorii suport contează; celelalte exemple de învățare sunt ignoreate.
- Găsește clasificatorul pentru care *marginea  $\rho$  e maximă*



Localizarea obiectelor la nivel de fereastră:  
metoda ferestrei glisante (sliding-window)

# Recunoașterea claselor de obiecte la nivel generic

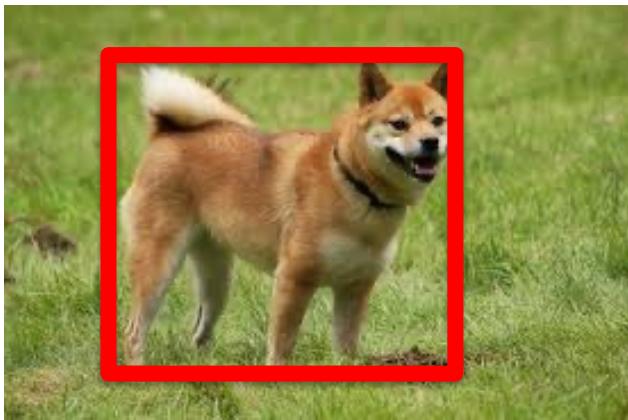
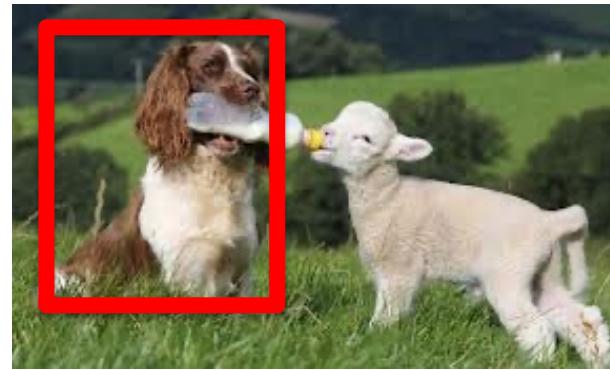
- Mașină



Vrem să recunoștem toate instanțierile unei clase de obiecte  
Clasificare – DA/NU. Localizare – UNDE?

# Recunoașterea claselor de obiecte la nivel generic

- Recunoaște orice câine în imagine



# Recunoașterea claselor de obiecte la nivel generic

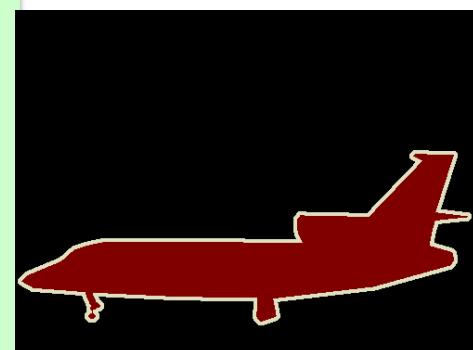
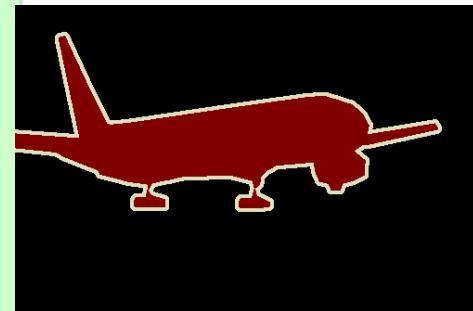
imagini inițiale



localizare - fereastră



localizare - contur



# Dificultăți: robustețe



Iluminare



Postura obiectului



Mărime



Mascarea  
obiectului



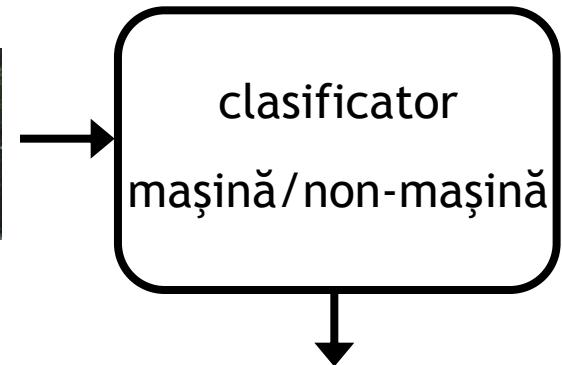
Variabilitate  
înfățisare intra-clasă



Poziționarea  
camerei

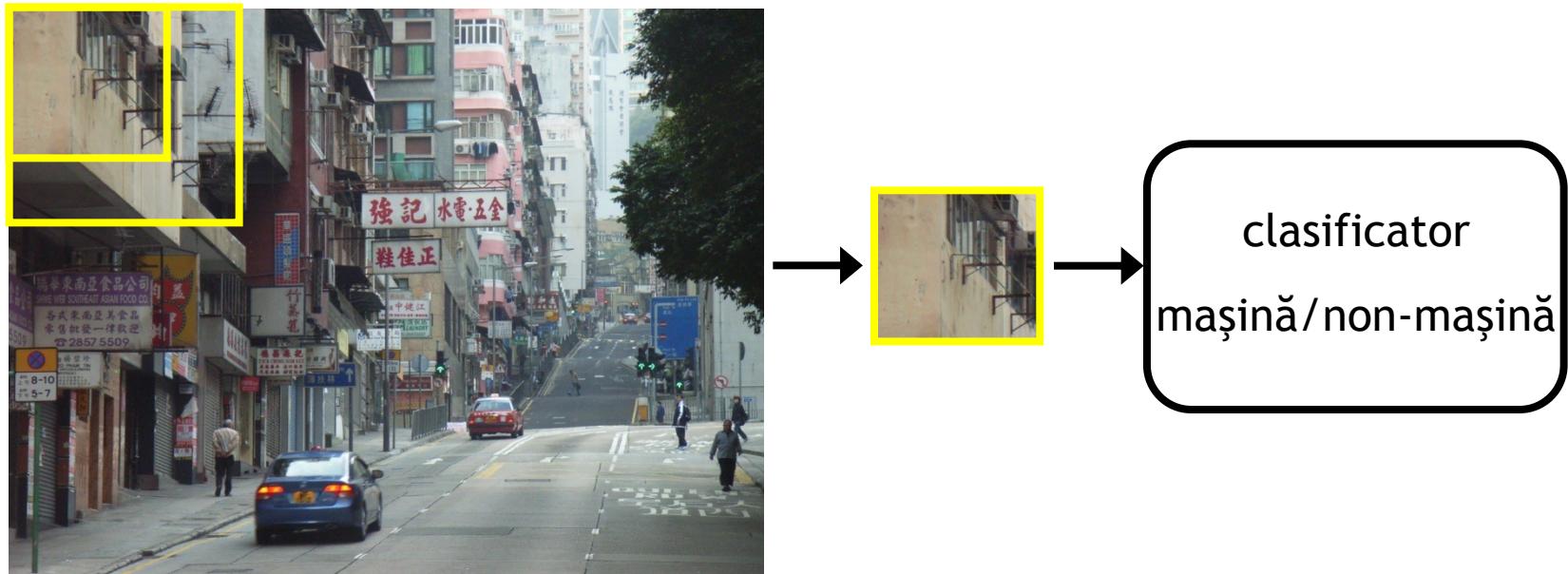
# Ideea principală: detectare via clasificare

Componența de bază: un clasificator binar



# Ideea principală: detectare via clasificare

Dacă obiectul este într-o scenă aglomerată, glisează o fereastră căutând obiectul.



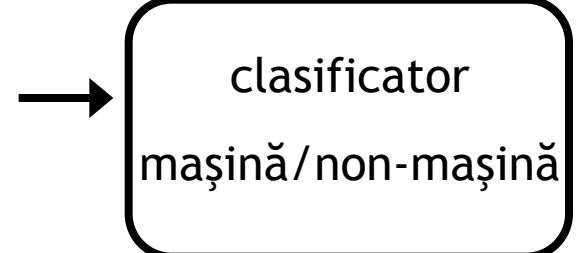
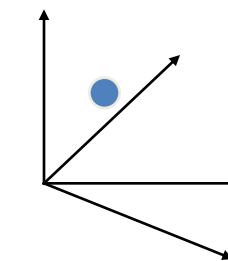
# Ideea principală: detectare via clasificare

Pași:

1. Obținem exemple de învățare (pozitive + negative)
2. Definim caracteristici
3. Definim clasificator



Exemple de învățare



Extragere de  
caracteristici

# Ideea principală: detectare via clasificare

- Considerăm toate ferestrele dintr-o imagine
  - consideră ferestre poziționate în fiecare pixel, de mărimi diferite (+ orientări diferite)
- Pentru fiecare fereastră decide:
  - “Contine sau nu această fereastră o instanță a clasei de obiecte X?”