

Concepțe și aplicații în Vederea Artificială

Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

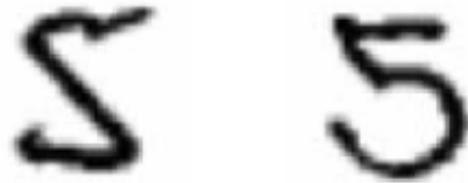
Radu Ionescu

radu.ionescu@fmi.unibuc.ro

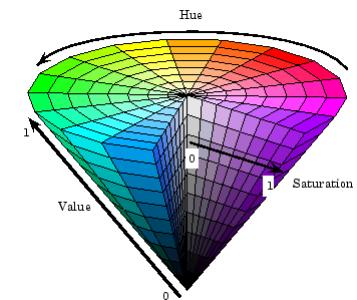
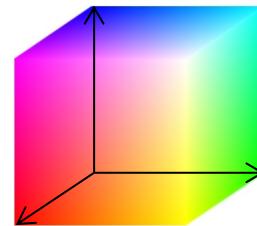
Curs optional
semestrul I, 2023-2024

Cursul trecut

- Compararea contururilor

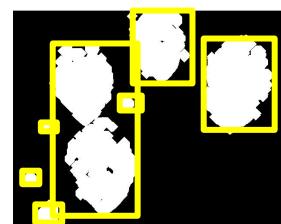
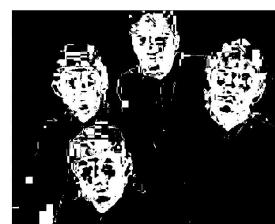


- Spațiile de culori RGB și HSV



- Imagini binare

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



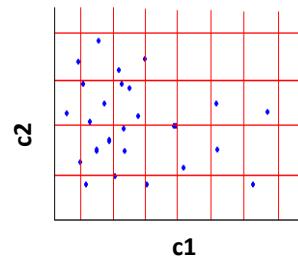
- Recunoașterea claselor de obiecte

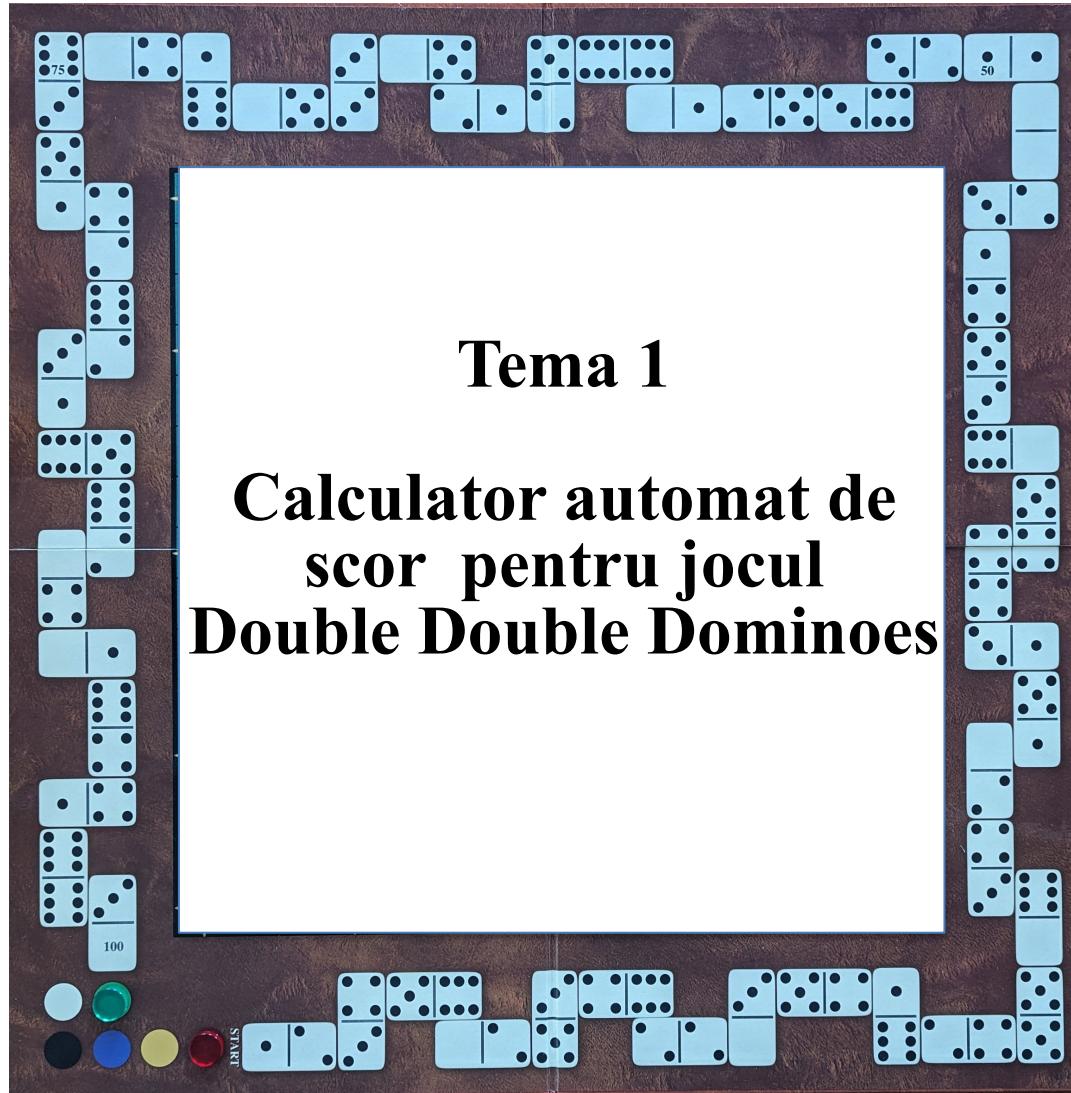
- privire de ansamblu
- ce este o clasă de obiecte?



Cursul de azi

- Tema 1: Calculator automat de scor pentru jocul Double Double Dominoes
- Clasificarea imaginilor
 - reprezentarea caracteristicilor prin histograme
- Localizarea claselor de obiecte
 - metoda ferestrei glisante





Tema 1

**Calculator automat de scor pentru jocul
Double Double Dominoes**

<https://tinyurl.com/CAVA-2023-TEMA1>

Double Double Dominoes

Double Double Dominoes (DDD) este o variantă a jocului de domino care îmbină idei din jocul de Scrabble cu cel de domino (Figura 1). Scopul jocului este de a obține puncte construind lanțuri de piese de domino care acoperă pătrate speciale de pe tablă. Pe măsură ce un jucător plasează aceste piese de domino pe poziții mai îndepărțate de centrul tablei crește șansa de a obține un punctaj mai mare. Punctajul obținut poate crește și prin utilizarea pieselor de domino duble (cu aceeași valoare la ambele capete, cum ar fi 0-0, 1-1, etc.) sau prin valorificarea poziției pionului plasat pe traseul pentru scor.

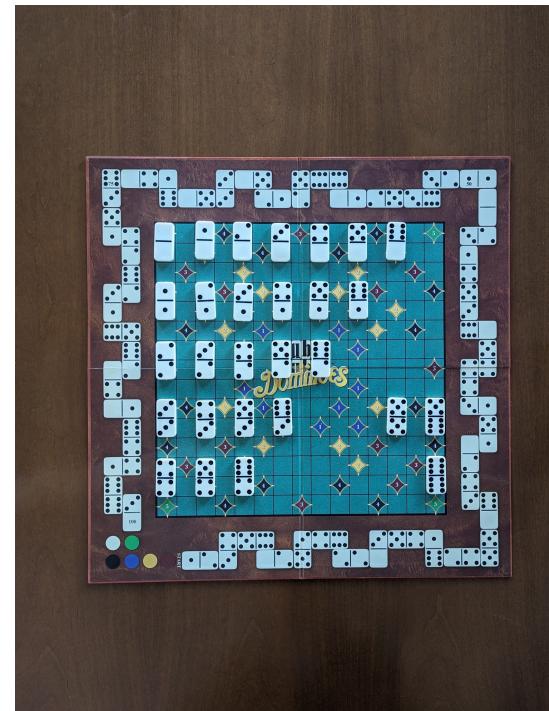
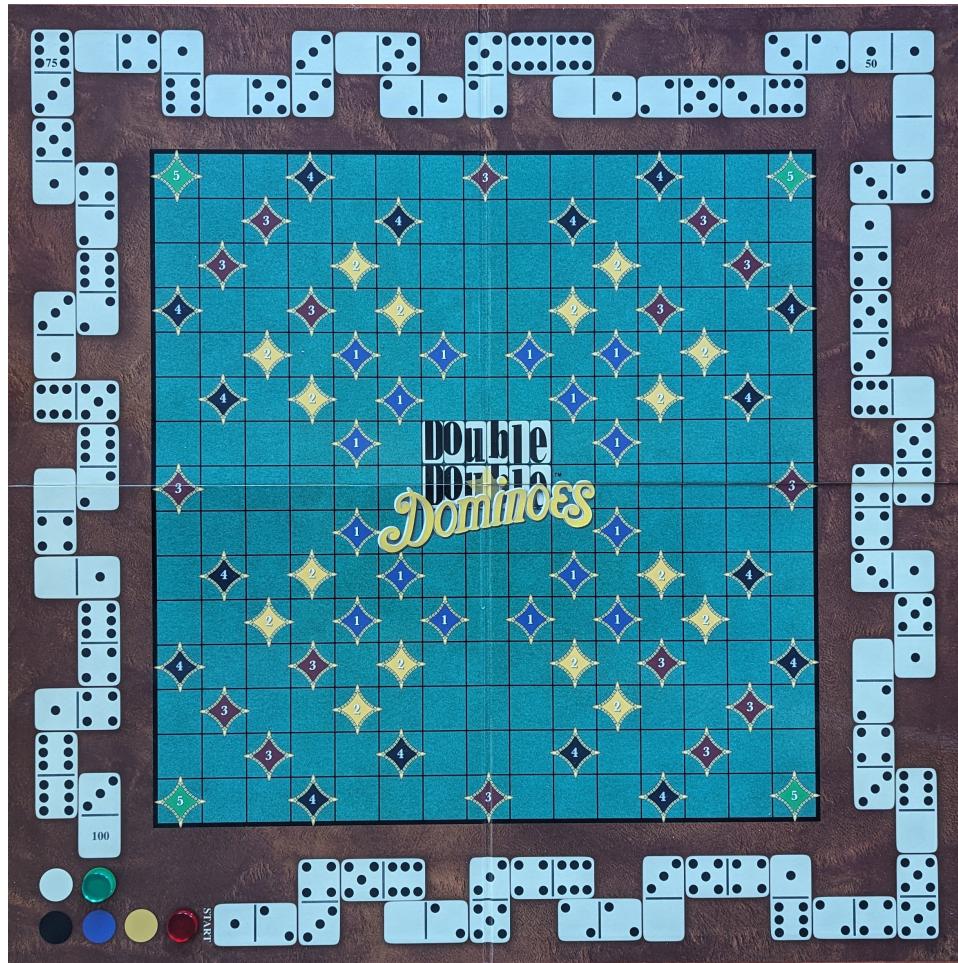


Figura 1: Tabla de joc DDD (stânga) și toate piesele posibile de domino plasate pe tablă (dreapta).

Tabla de joc

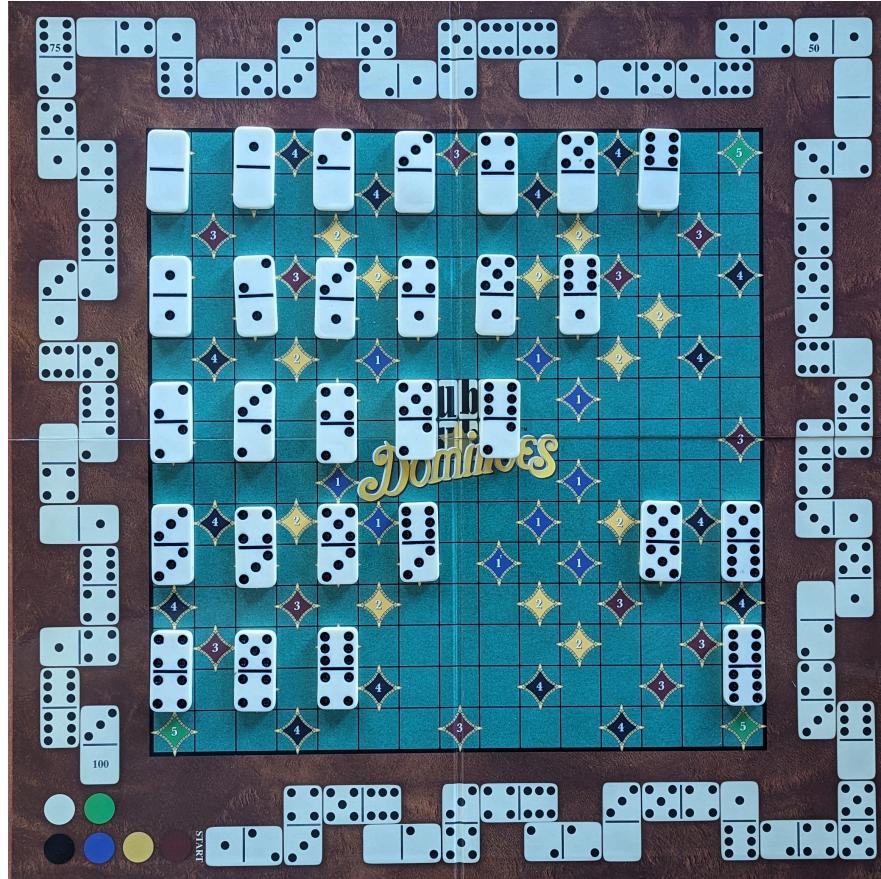
Tabla de joc pentru DDD (Figura 1) este împărțită într-un careu de 15×15 pătrate și are un traseu pentru scor ce ocupă marginea tablei de joc. Pe acest traseu se vor plasa pionii jucătorilor. Tabla este marcată cu pătrate speciale care conțin *romburi* colorate ce au asociat un punctaj:

- 16 romburi albastre pentru 1 punct;
- 16 romburi galbene pentru 2 puncte;
- 16 romburi maro pentru 3 puncte;
- 16 romburi negre pentru 4 puncte;
- 4 romburi verzi pentru 5 puncte.



Piese de joc

O piesă de domino (prescurtat *domino*) are două capete, iar fiecare capăt are 7 valori posibile, date de numărul de puncte negre, de la 0 la 6. Cele două capete sunt separate de o linie neagră, verticală sau orizontală, în funcție de cum e poziționat domino-ul pe tablă. Există în total 28 de piese de domino diferite (0-0, 0-1, ..., 6-6). Acestea sunt plasate pe tablă în poziție verticală în Figura 1 în partea dreaptă.

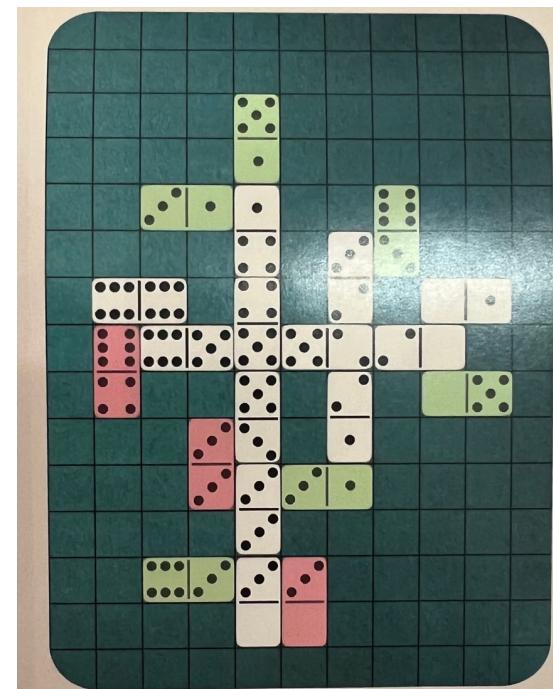
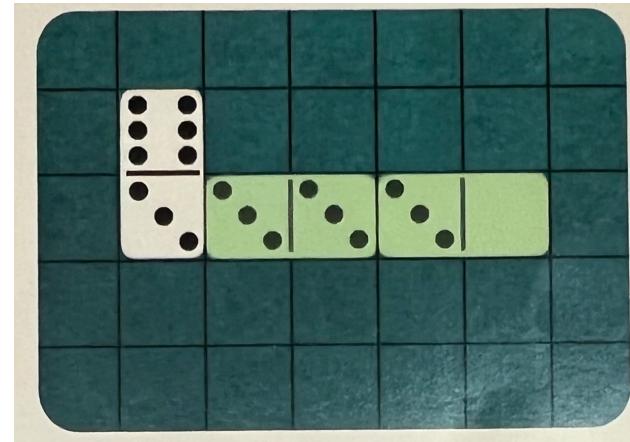
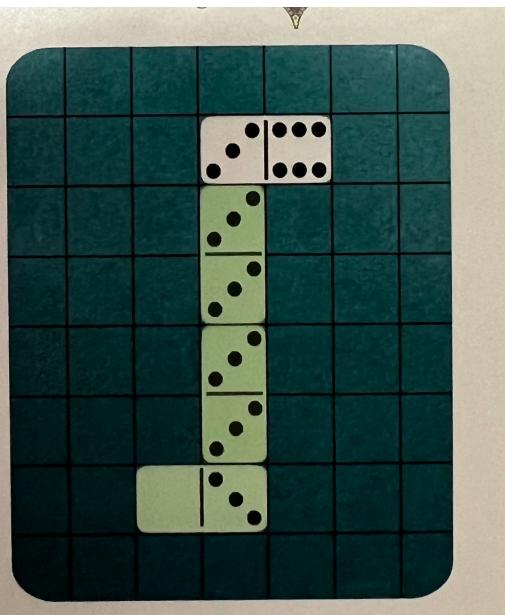


Sacul cu piese de domino

Pentru acest joc, există un sac cu piese de domino care conține fiecare domino repetat de două ori, în total sunt 56 de piese de domino. Fiecare jucător trage la începutul jocului 3 domino-uri din sac. Când îi vine rândul, jucătorul curent extrage un domino din sac (are astfel 4 domino-uri în mână) iar după ce plasează unul sau mai multe domino-uri pe tablă trage din sac domino-uri pentru a își completa numărul de domino-uri astfel încât să aibă mereu 3, cât timp sacul mai conține piese. Dacă nu plasează niciun domino pe tablă, atunci va elimina unul dintre cele 4 domino-uri din mână pentru a rămâne cu 3. În cele ce urmează, vom ignora acțiunile de a extrage sau de a elmina domino-uri, informația este prezentată pentru o înțelegere completă a jocului.

Desfășurarea jocului

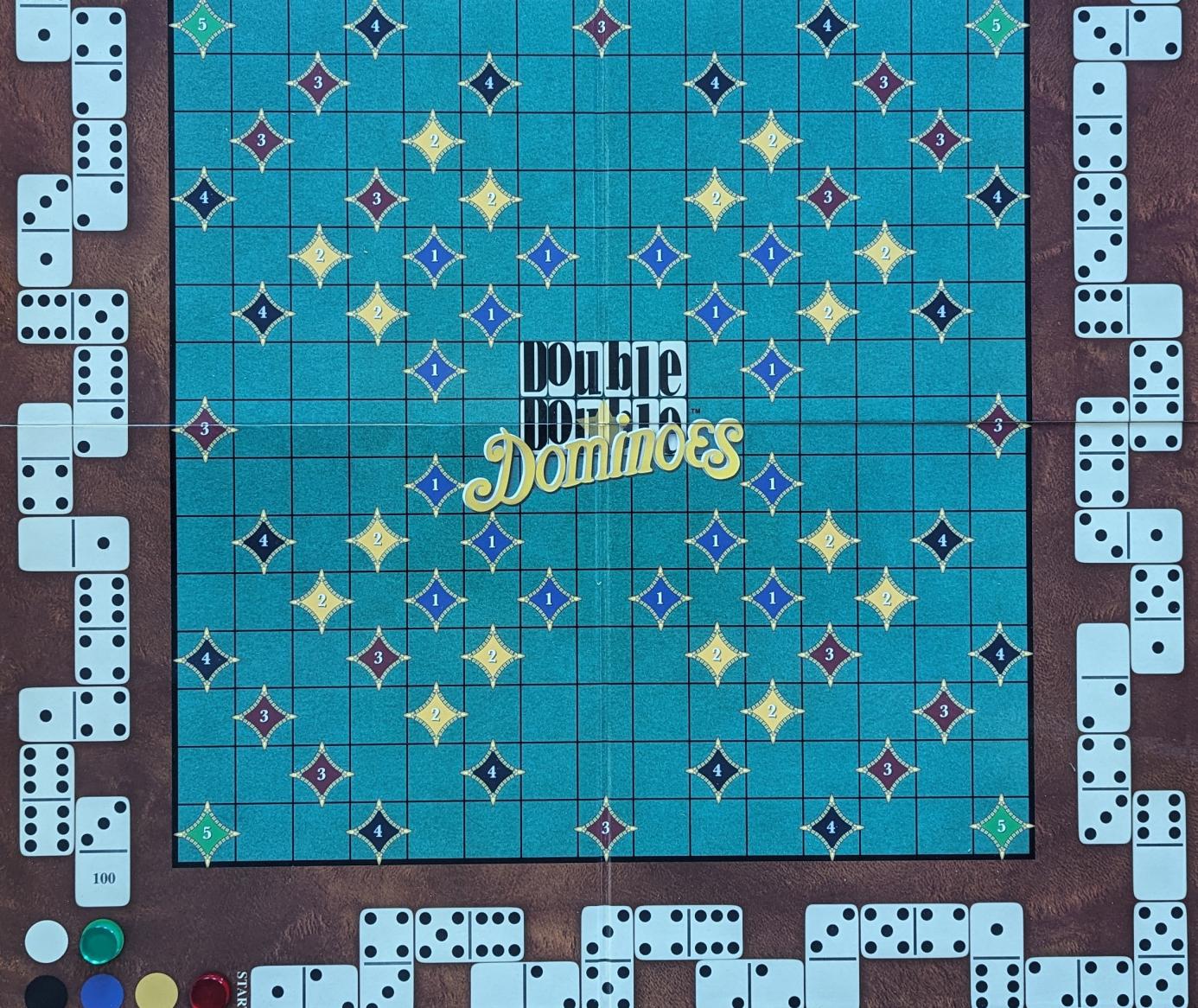
Jucătorul ce începe jocul trebuie să plaseze un domino pe pătratul marcat cu stea \star din centrul tablei. Fiecare alt domino trebuie plasat astfel încât un capăt al său să atingă exact un domino cu numărul corespunzător. Celălalt capăt este fie liber (nu atinge nicio altă piesă), fie atinge un alt domino cu care se potrivește (are același număr). Atunci când un jucător pune un domino dublu, acesta poate alege să plaseze imediat un alt domino pe tabla de joc, dar acesta trebuie plasat legat de acel domino dublu. Rândul jucătorului se încheie atunci când acesta decide să nu mai plaseze domino-uri sau când plasează un domino care nu este dublu. Figura 2 arată configurații valide și invalide pe o tablă de joc de domino.



Calcularea punctajului

Traseul pentru scor ocupă marginea tablei de joc. De fiecare dată când un jucător obține puncte, trebuie să mute imediat pionul său pe traseu cu un număr de căsuțe egal cu numărul de puncte obținute. Există două modalități de a obține puncte:

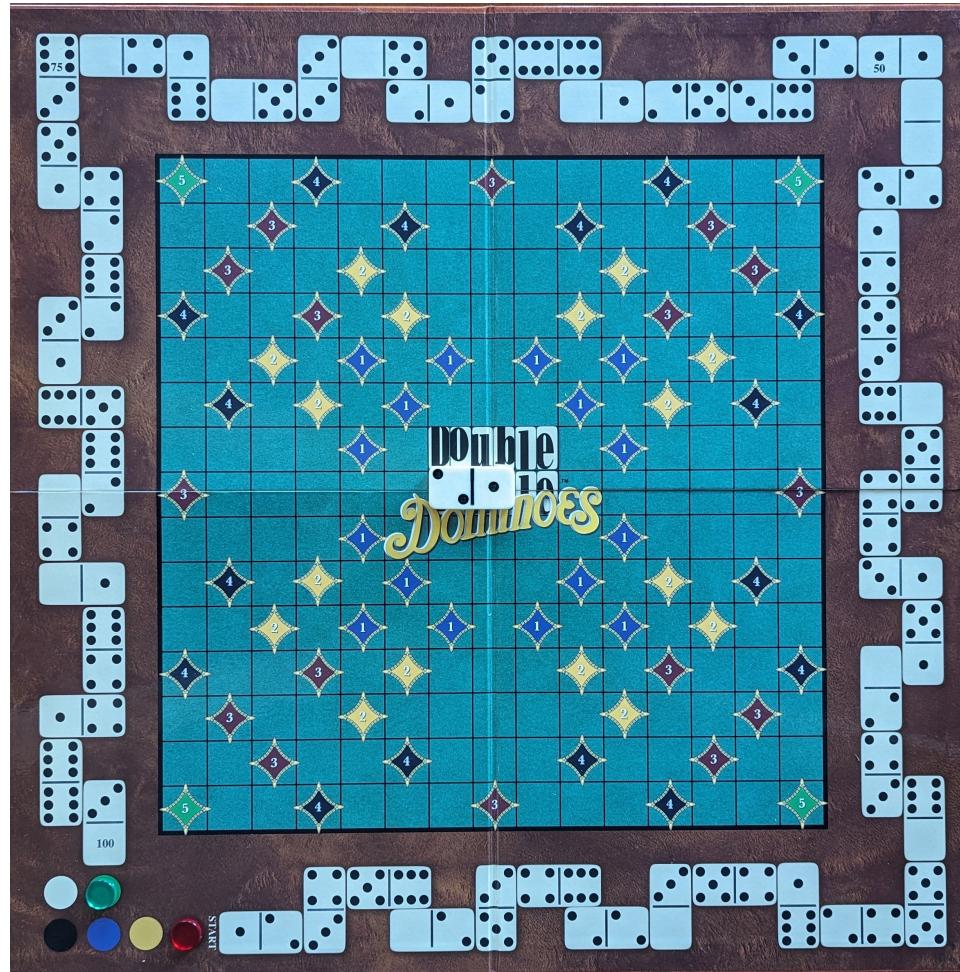
- *prin plasarea unui domino pe pătratele speciale* (ce conțin romburile colorate cu punctaj asociat) de pe tablă: atunci când un jucător pune un domino care acoperă un pătrat special, jucătorul obține punctajul asociat indicat de numărul din romb. Dacă jucătorul joacă un domino dublu, atunci acesta obține dublul punctelor.
- *prin plasarea unui domino ce are unul dintre cele două capete egal în valoare cu căsuța pionului pe traseul de scor*: de fiecare dată când un domino este plasat pe tablă iar unul din numerele de pe acesta se potrivește cu numărul căsuței pe care se află un pion de pe traseu, jucătorul care are acel pion primește 3 puncte bonus. De exemplu, dacă pionul se află pe o căsuță cu valoarea 4, iar pe tablă se plasează un domino ce are unul din capete inscripționat cu numărul 4, pionul respectiv se va muta 3 căsuțe înainte. Punctele bonus se acordă doar o singură dată pentru fiecare domino jucat.



Example de mutări și calculare a scorului

Verdele mută (dominoul 2-1 este jucat, nu se întâmplă nimic)

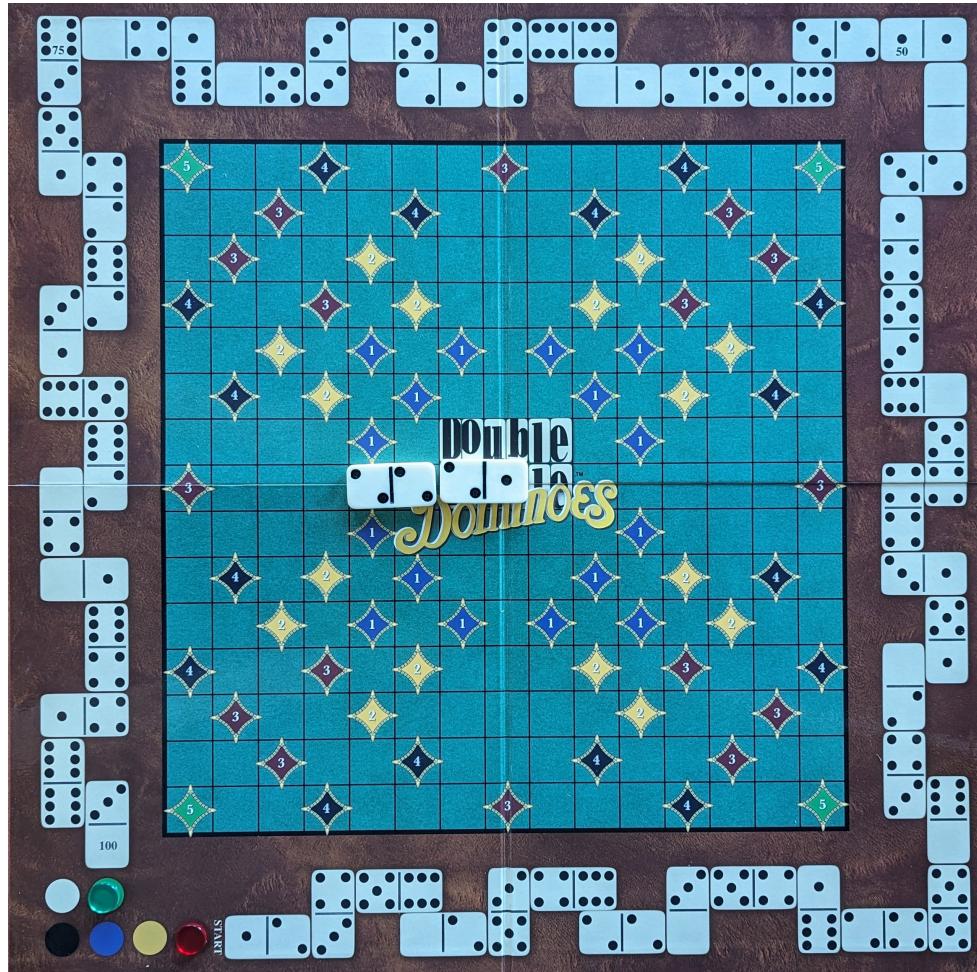
Scorul după mutare: **Verde 0 - 0 Roșu**



Example de mutări și calculare a scorului

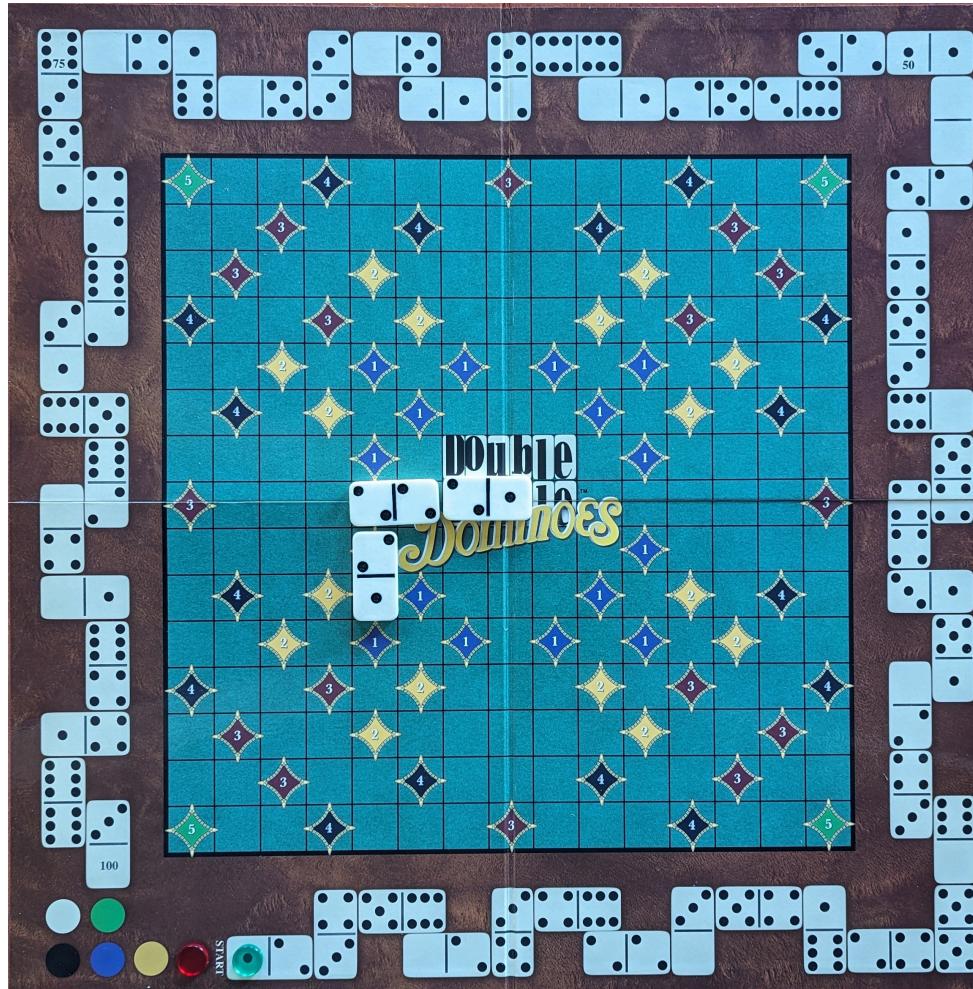
Roșu mută (dominoul 2-2 este jucat, nu se întâmplă nimic)

Scorul după mutare: **Verde 0 - 0 Roșu**



Example de mutări și calculare a scorului

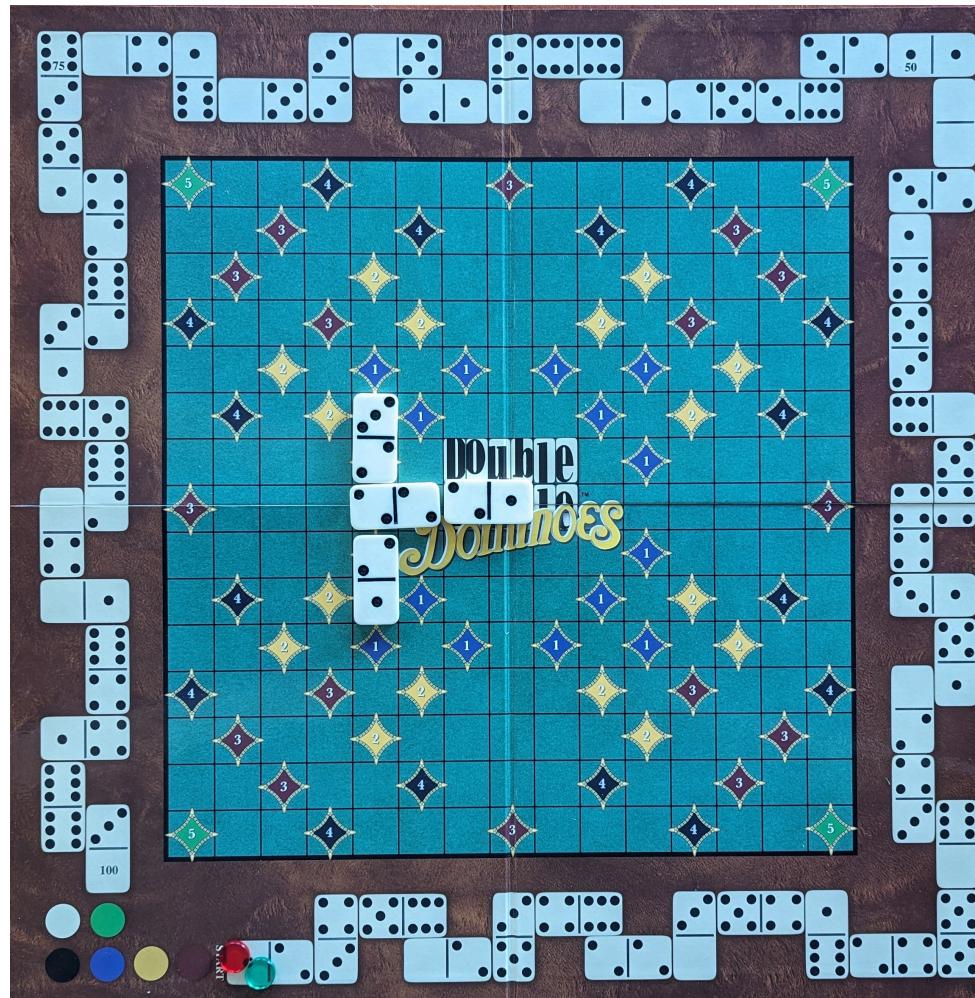
Verdele mută (dominoul 2-1 este jucat pe rombul albastru – 1p)
Scorul după mutare: **Verde 1 - 0 Roșu**



Example de mutări și calculare a scorului

Roșu mută (dominoul 3-2 este jucat pe rombul albastru – 1p)

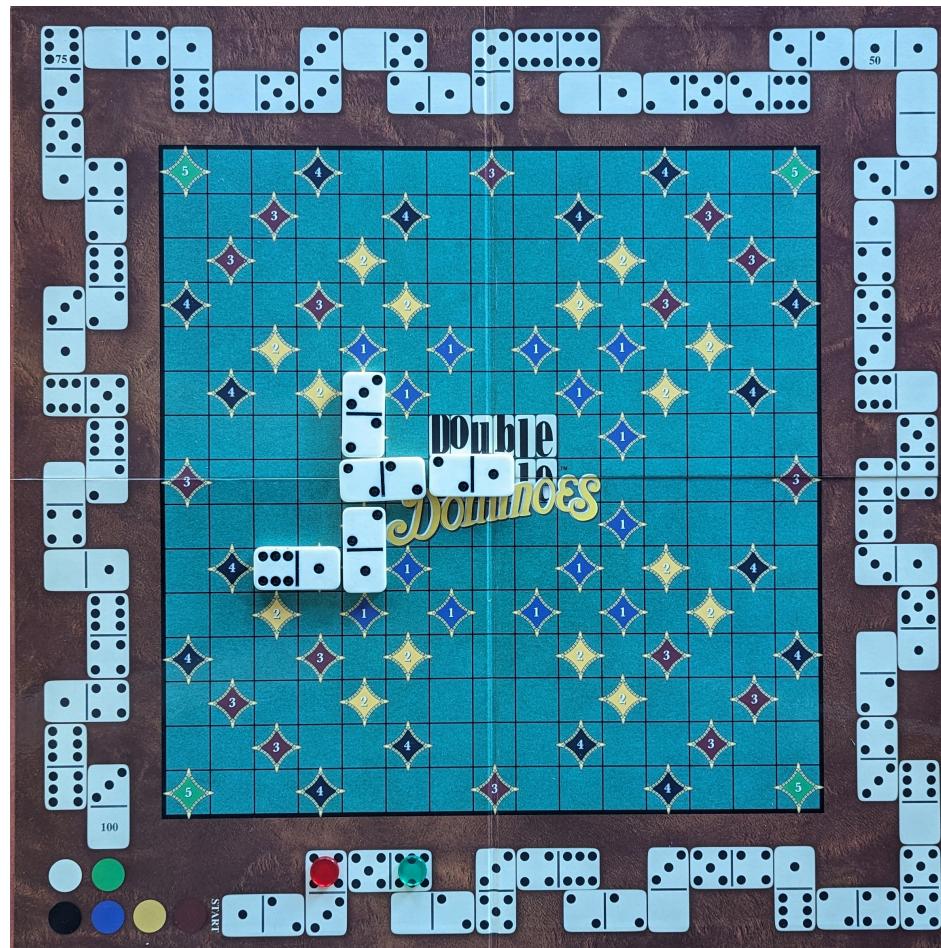
Scorul după mutare: **Verde 1 - 1 Roșu**



Example de mutări și calculare a scorului

Verdele mută (dominoul 6-1 este jucat pe rombul galben – 2p + bonus 3p pentru ambii jucători)

Scorul după mutare: **Verde 6 - 4 Roșu**

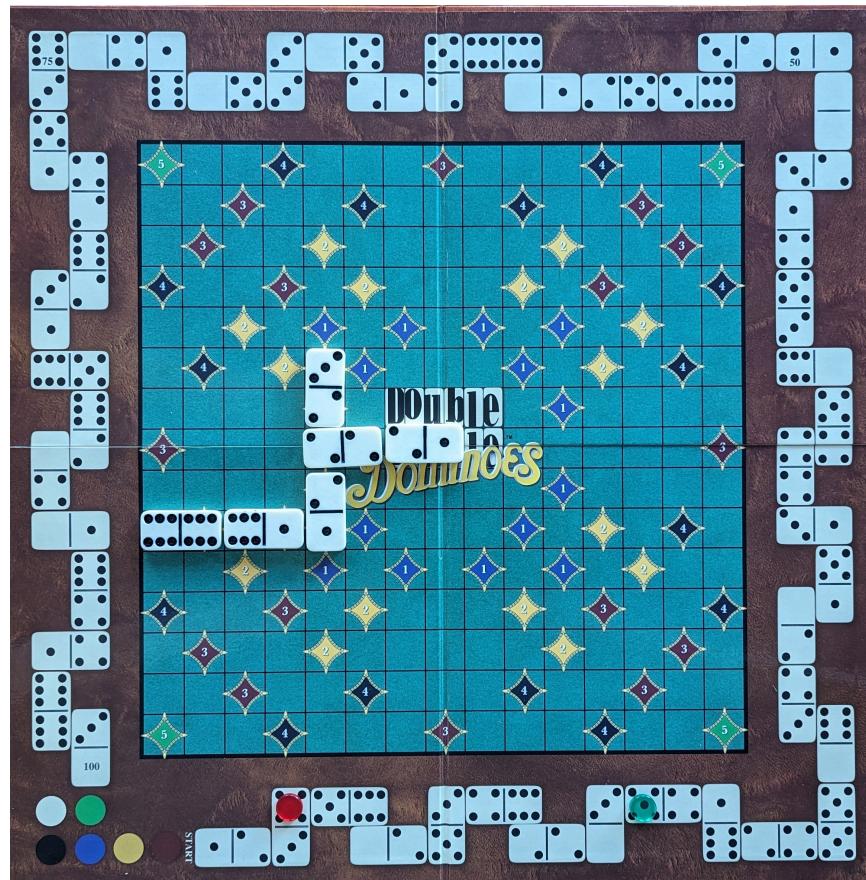


Example de mutări și calculare a scorului

Roșu cedează rândul (nu se întâmplă nimic)

Verdele mută (dominoul 6-6 este jucat pe rombul negru – $4p \times 2 +$ bonus 3p pentru jucătorul verde)

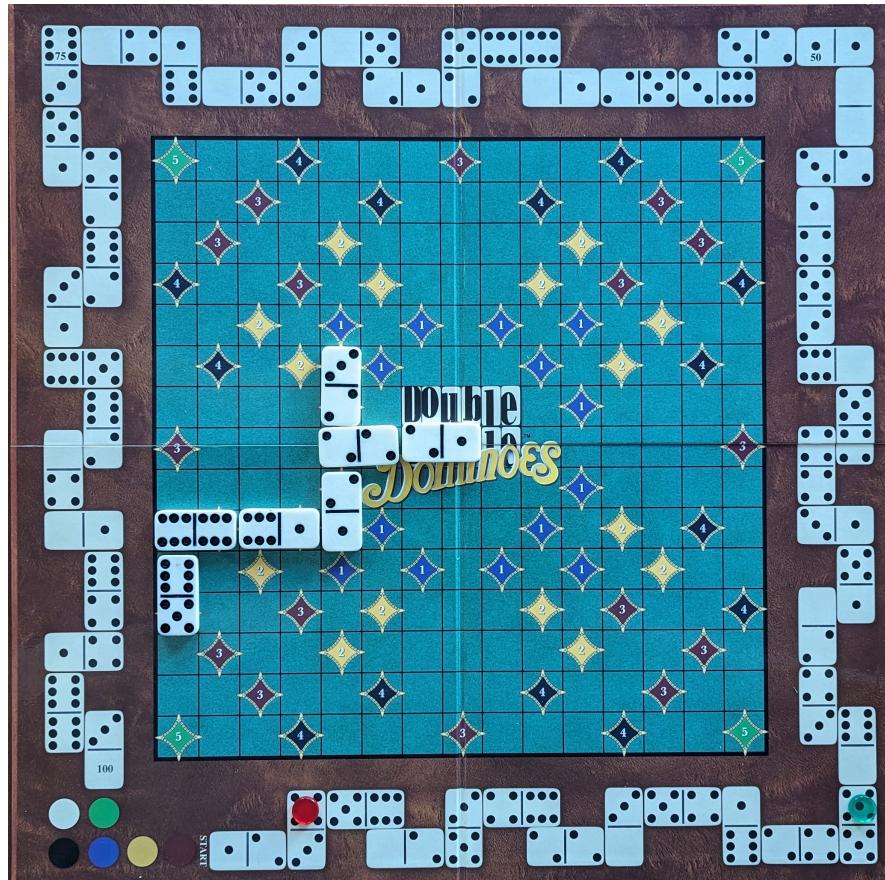
Scorul după mutare: **Verde 17 - 4 Roșu**



Example de mutări și calculare a scorului

Verdele mută (dominoul 6-5 este jucat pe rombul negru – 4p + bonus 3p pentru jucătorul verde)

Scorul după mutare: **Verde 24 - 4 Roșu**



Descrierea datelor

Arhiva cu materiale (disponibilă aici <https://tinyurl.com/CAVA-2023-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 Double Double Dominoes și cu piesele de domino 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 20 de mutări pentru 5 jocuri de DDD. În total sunt 100 de imagini de antrenare și 100 de fișiere cu adnotări corespunzătoare fiecărei imagini. Imaginea de antrenare i din jocul j este denumită ' $j_i.jpg$ ', unde $j \in \{1, 2, 3, 4, 5\}$ și $i \in \{01, 02, 03, \dots, 20\}$. Fișierul cu adnotări corespunzător are același format, extensia ' $.jpg$ ' fiind înlocuită de extensia ' $.txt$ '. Pentru fiecare joc, există un fișier $.txt$ ce conține informații despre ce jucător a efectuat mutarea în imaginea respectivă ($j_mutari.txt$ unde j este numărul jocului).

Descrierea datelor

 CAVA-2023-TEMA1	 >	
 antrenare	 >	
 evaluare	 >	
 imagini_auxiliare	 >	
 README.txt	 >	
 Tema1.pdf	 >	
 testare	 >	

Descrierea datelor

 CAVA-2023-TEMA1	 >	
 antrenare	 >	 01.jpg 
 evaluare	 >	 02.jpg 
 imagini_auxiliare	 >	 03.jpg 
 README.txt	 >	
 Tema1.pdf		
 testare	 >	

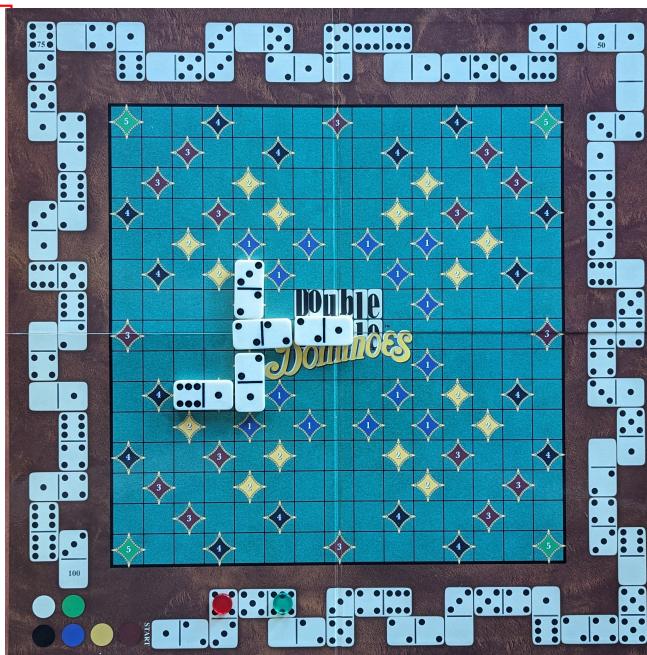
Descrierea datelor

CAVA-2023-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		✓
		1_09.txt		✓
		1_10.jpg		✓
		1_10.txt		✓
		1_11.jpg		✓
		1_11.txt		✓
		1_12.jpg		✓
		1_12.txt		✓
		1_13.jpg		✓

Adnotări

Fișierele cu adnotări conțin următoarele informații:

- poziția piesei de domino adăugată pe tablă, poziție numerotată de la stânga la dreapta și de sus în jos. Folosim ca reguli de adnotare regulile oficiale de Scrabble, astfel pentru a specifica o poziție pe tablă folosim numere de la 1 la 15 pentru linii și literele de la A la O pentru coloane.
- numerele corespunzătoare celor două capete ale fiecărei piese de domino adăugate pe tablă.
- scorul obținut în runda respectivă de jucătorul la mutare.



10C 6
10D 1
5

Evaluare

Directorul *evaluare* vă indică cum să vă scrieți codul astfel încât să respectați formatul fișierelor cu rezultate impus pentru 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, 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.
- *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 (ground-truth) vor fi disponibile după faza a doua.

Evaluare

CAVA-2023-TEMA1 ✓ >	antrenare ✓ > evaluare ✓ > imagini_auxiliare ✓ > README.txt ✓ Tema1.pdf ✓ testare ✓ >	cod_evaluare ✓ > fake_test ✓ > fisiere_solutie ✓ >	evaluateaza_solutie.py
---	--	--	------------------------

Evaluare

 CAVA-2023-TEMA1 	 antrenare   evaluare   imagini_auxiliare   README.txt   Tema1.pdf   testare 	 cod_evaluare   fake_test   fisiere_solutie 	 1_01.jpg  1_02.jpg  1_03.jpg  1_04.jpg  1_05.jpg  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  1_18.jpg  1_19.jpg  1_20.jpg  1_mutari.txt
---	--	--	--

Evaluare

CAVA-2023-TEMA1	antrenare	cod_evaluare	331_Alexe_Bogdan	1_01.txt
	evaluare	fake_test		1_02.txt
	imagini_auxiliare	fisiere_solutie		1_03.txt
	README.txt			1_04.txt
	Tema1.pdf			1_05.txt
	testare			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
				1_18.txt
				1_19.txt
				1_20.txt

Diferențe între antrenare și testare

Foarte important. Există o diferență între datele de antrenare și cele de testare: doar în imaginile de antrenare aveți pionii plasați pe traseu, în timp ce în imaginile de testare pionii nu vor fi prezenți (de fapt, este sarcina voastră să descoperiți poziția lor). Figura 5 exemplifică diferența între imaginile de antrenare și cele de testare.

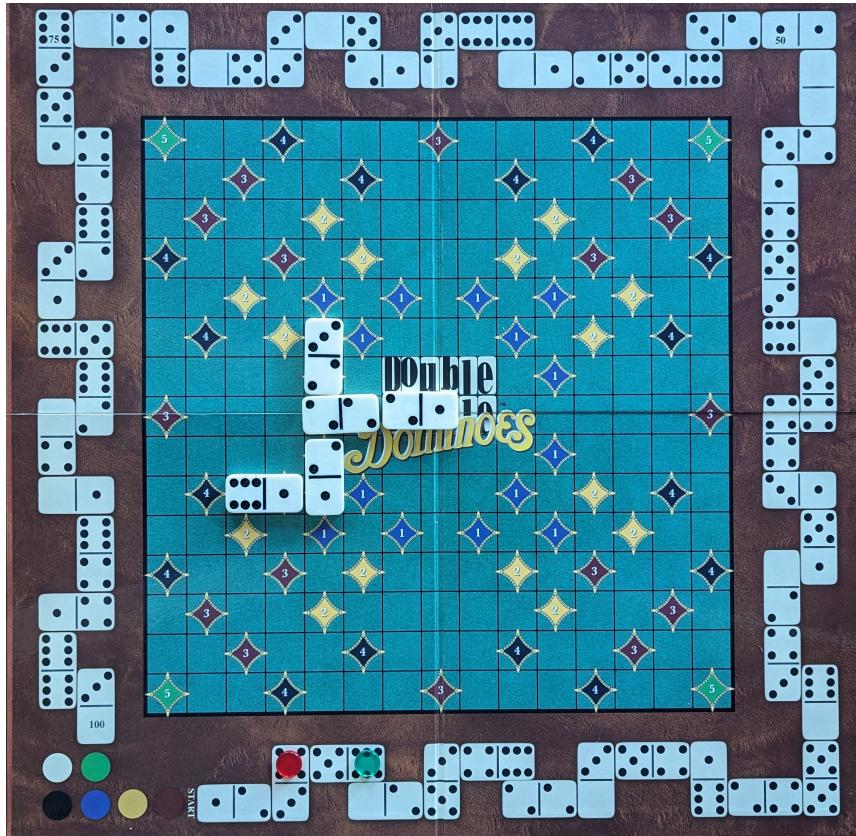


Figura 5: Imagine de antrenare cu pioni(stânga), imagine de testare fără pioni(dreapta).

Cerințe și notare

Scriți un program în Python/Jupyter notebook care extrage automat informațiile vizuale dintr-o anumită rundă dintr-un joc de Double Double Dominoes și calculează scorul pentru runda respectivă. Distribuția datelor de testare este aceeași ca la antrenare, imaginile fiind luate în aceleași condiții.

- **Task 1 - 5 puncte** - vom evalua performanța algoritmului vostru pe o mulțime de 100 de imagini de testare ce reprezintă 5 jocuri de DDD, fiecare joc având 20 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 (ambele capete). Nu există punctaje parțiale. Fiecare configurație corectă valorează 0.05 puncte pentru un total de **5 puncte**;
- **Task 2 - 2 puncte** - vom evalua performanța algoritmului vostru pe aceleași 100 de imagini de la Task-ul 1, pentru piesa plasată trebuie să recunoașteți numerele care se află pe piesa respectivă (pe cele două capete). Veți primi punctajul numai dacă algoritmul vostru recunoaște corect ambele numere. Nu există punctaje parțiale. Fiecare configurație corectă a numerelor valorează 0.02 puncte pentru un total de **2 puncte**;
- **Task 3 - 2 puncte** - vom evalua performanța algoritmului vostru pe aceleași 100 de imagini de la Task-urile 1 și 2, pentru fiecare imagine trebuie să calculați scorul obținut de jucătorul curent care realizează la runda respectivă. Veți primi punctajul numai dacă algoritmul vostru calculează corect scorul pentru runda curentă. Nu există punctaje parțiale. Pentru fiecare scor corect calculat primiți 0.02 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 marți, 5 decembrie, ora 23:59 la link-ul acesta <https://tinyurl.com/CAVA-2023-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, 6 decembrie, vom publica datele de test în directorul *testare* de la adresa <https://tinyurl.com/CAVA-2023-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-2023-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

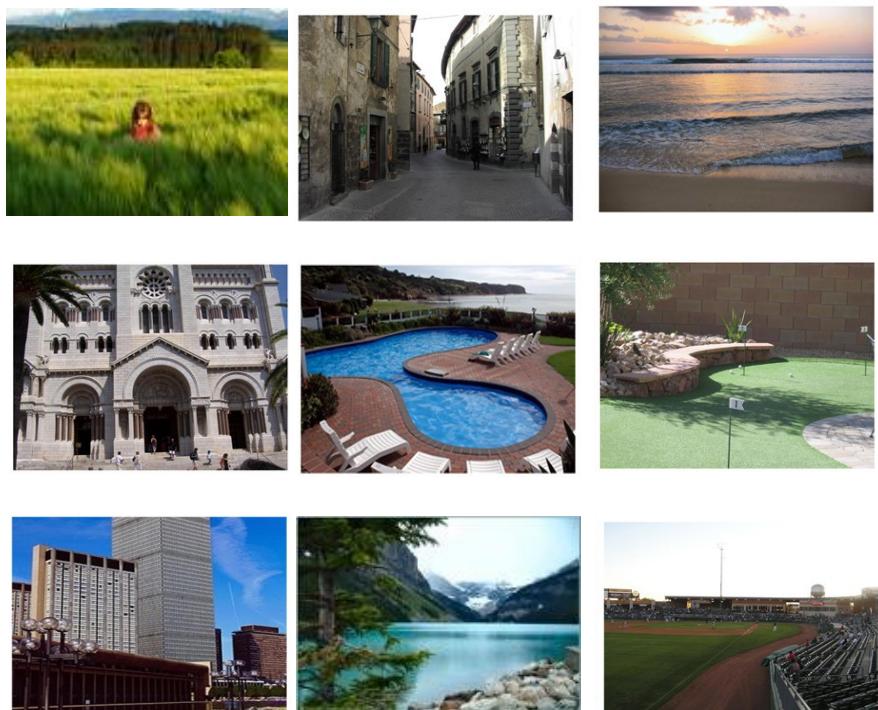


Alt exemplu: 2 clase - indoor/outdoor

Indoor – exemple



Outdoor - exemple



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

outdoor



outdoor

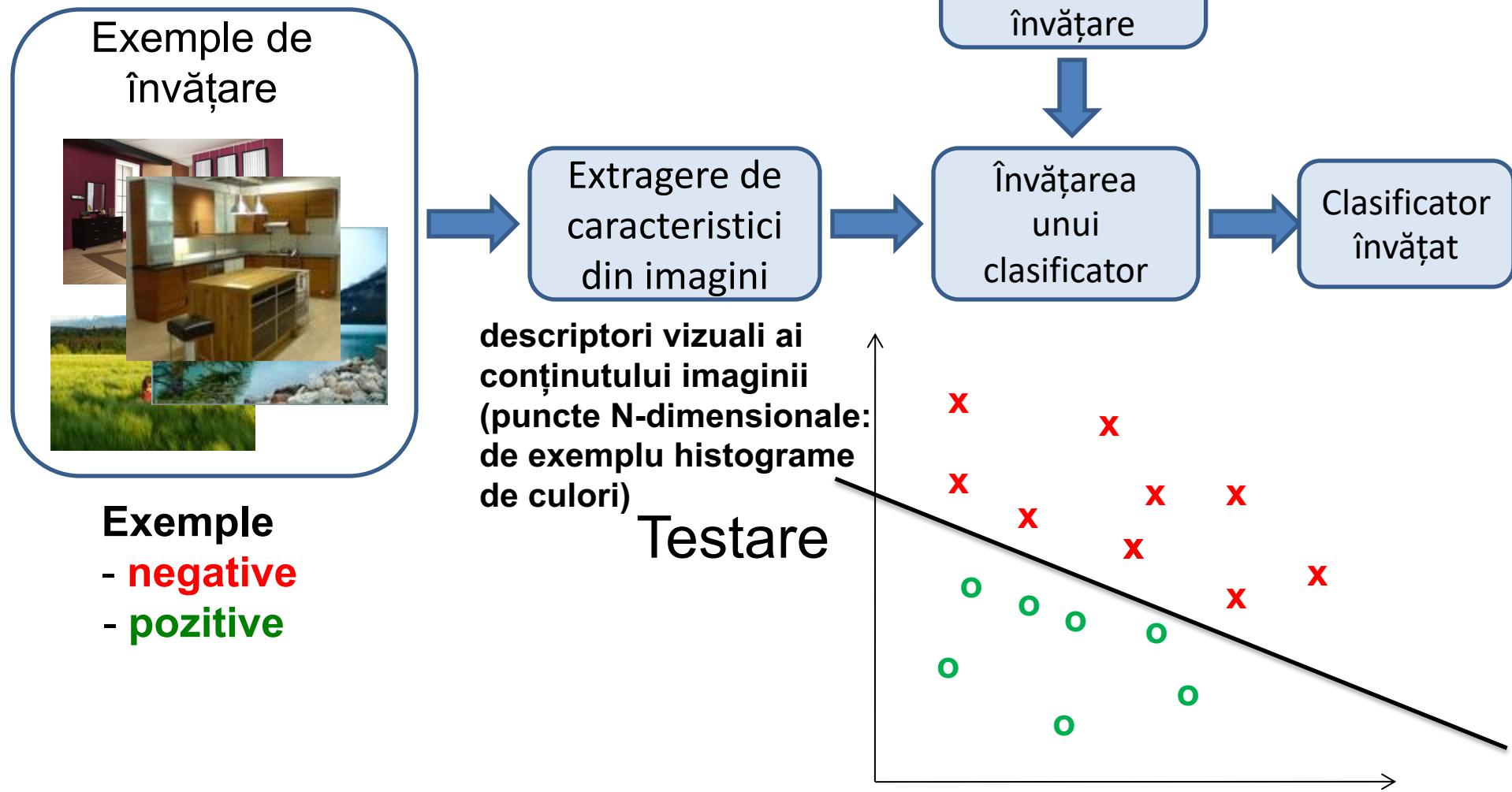


indoor

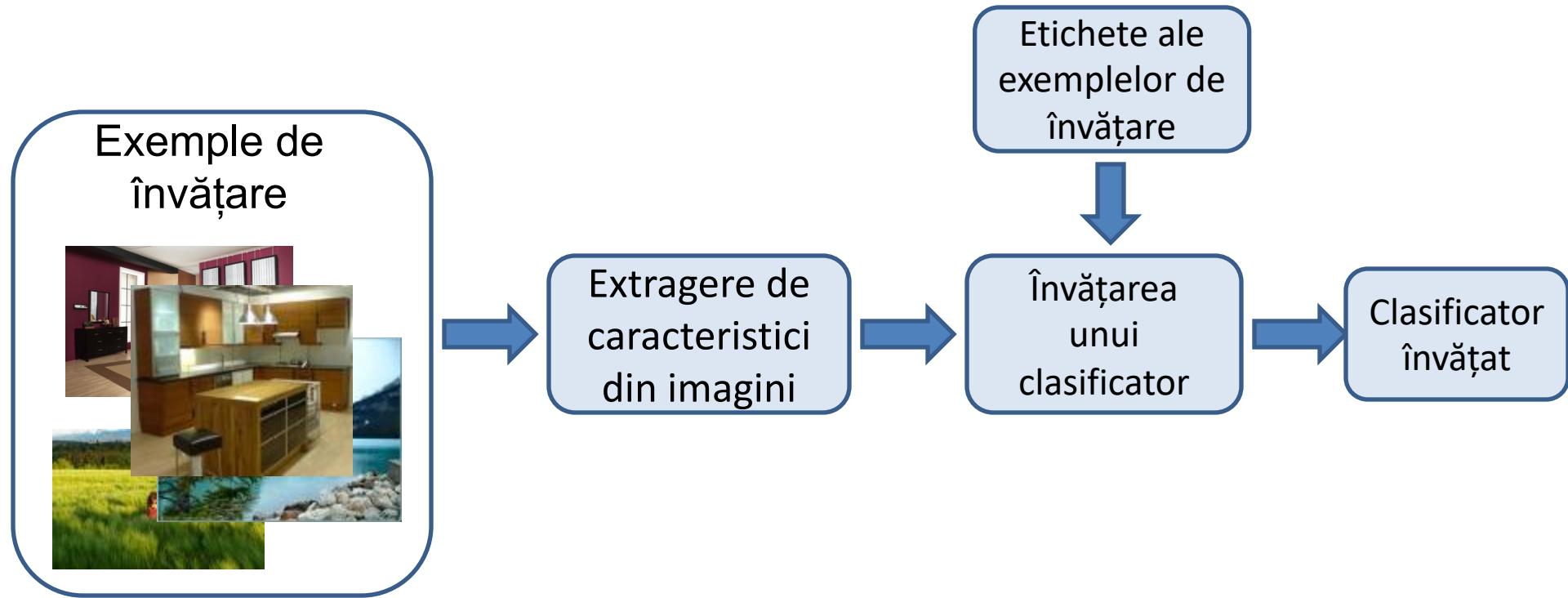


Imagini test:

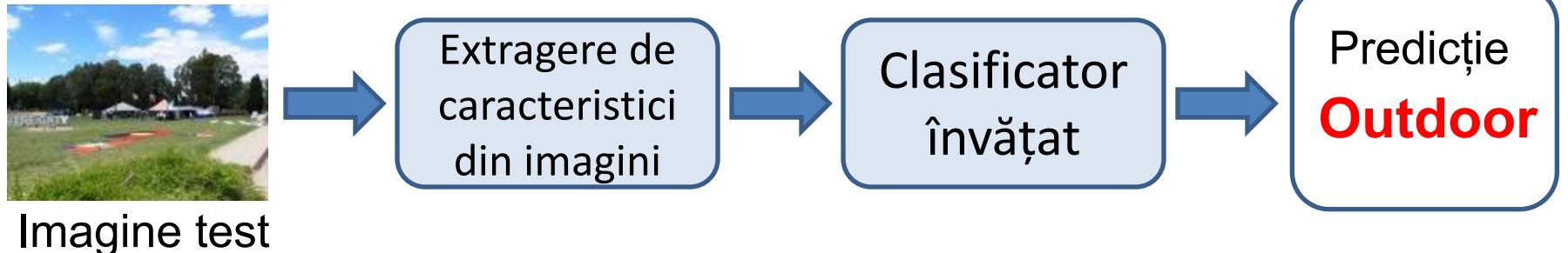
Învățarea supervizată



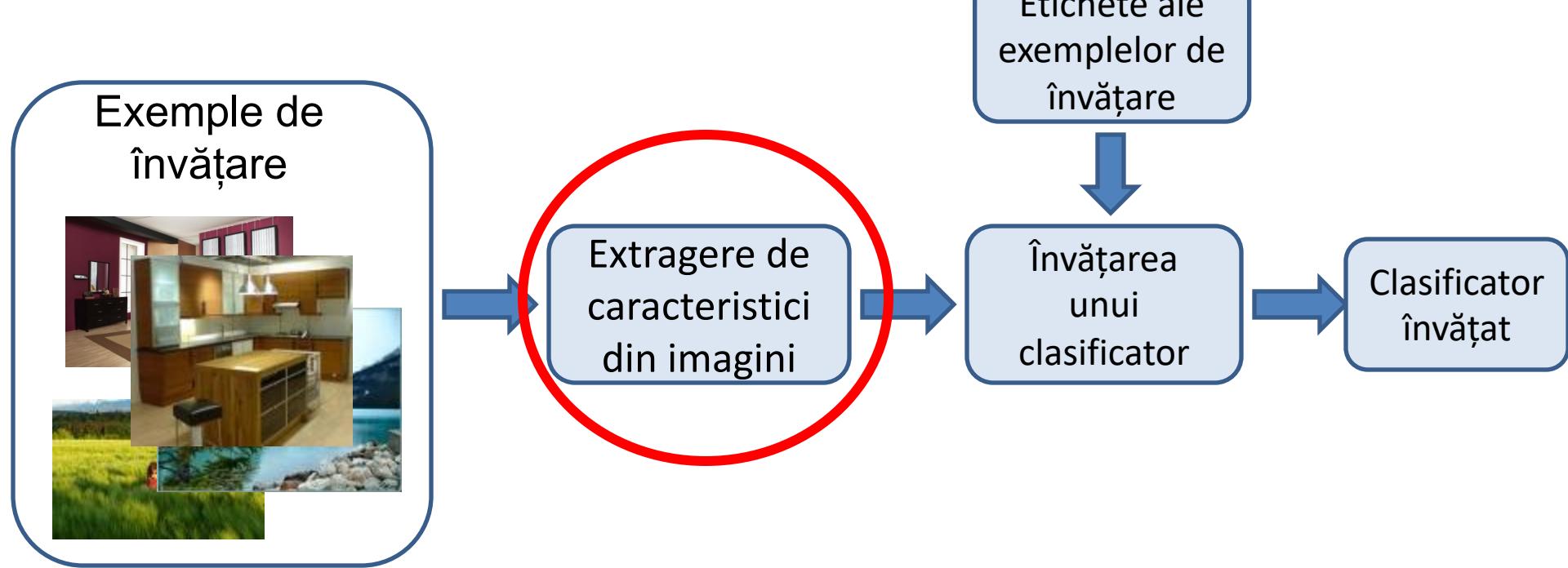
Etapa de testare



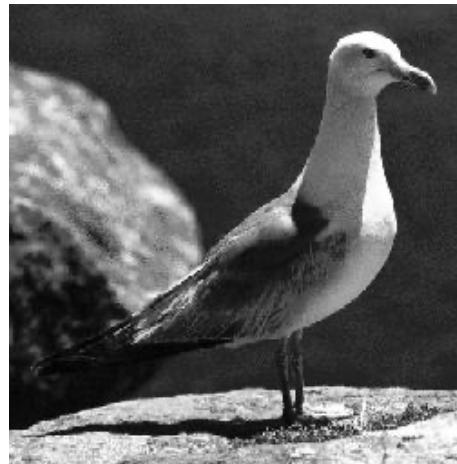
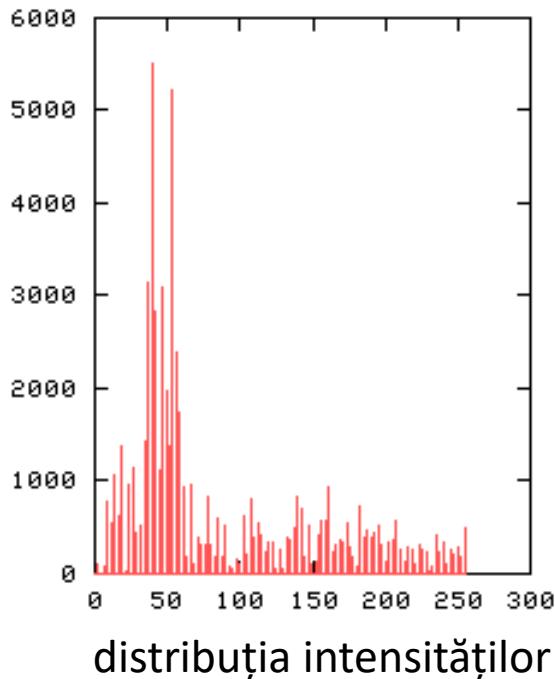
Testare



Extragere de caracteristici din imagini



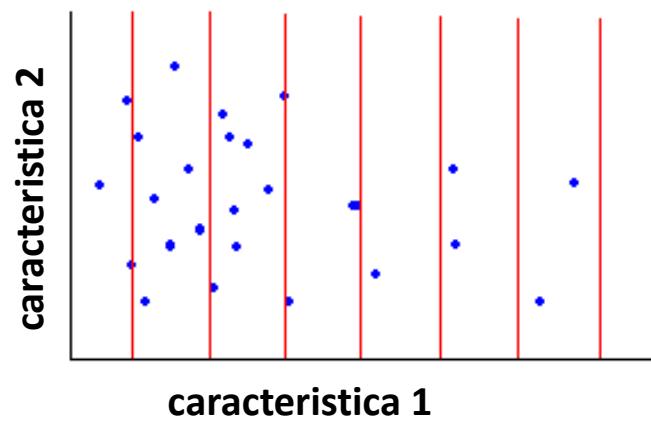
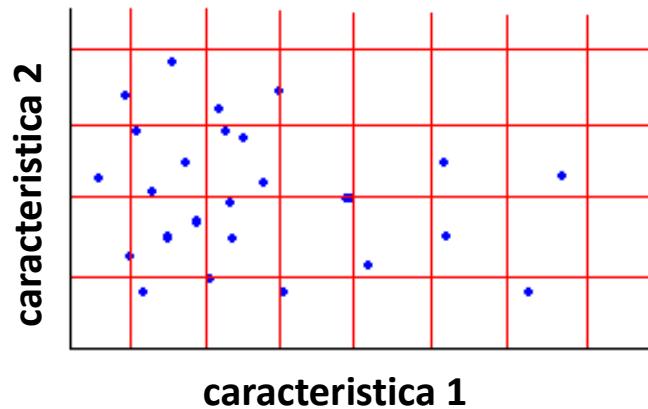
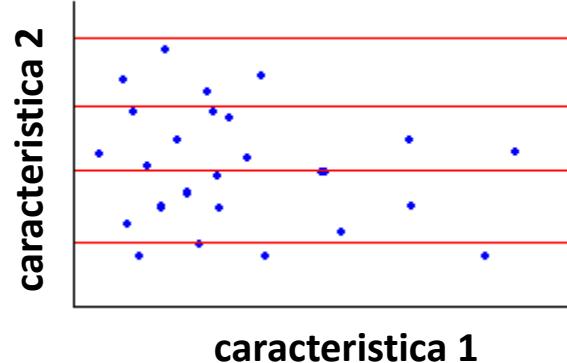
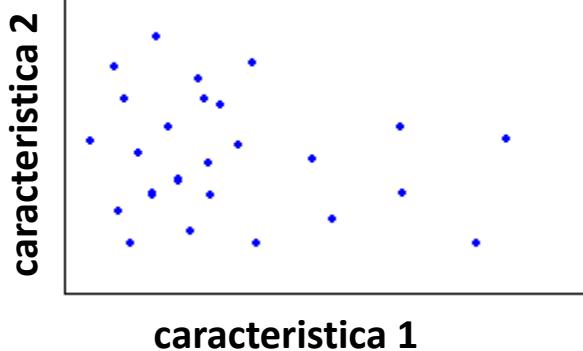
Histograme pentru o caracteristică



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



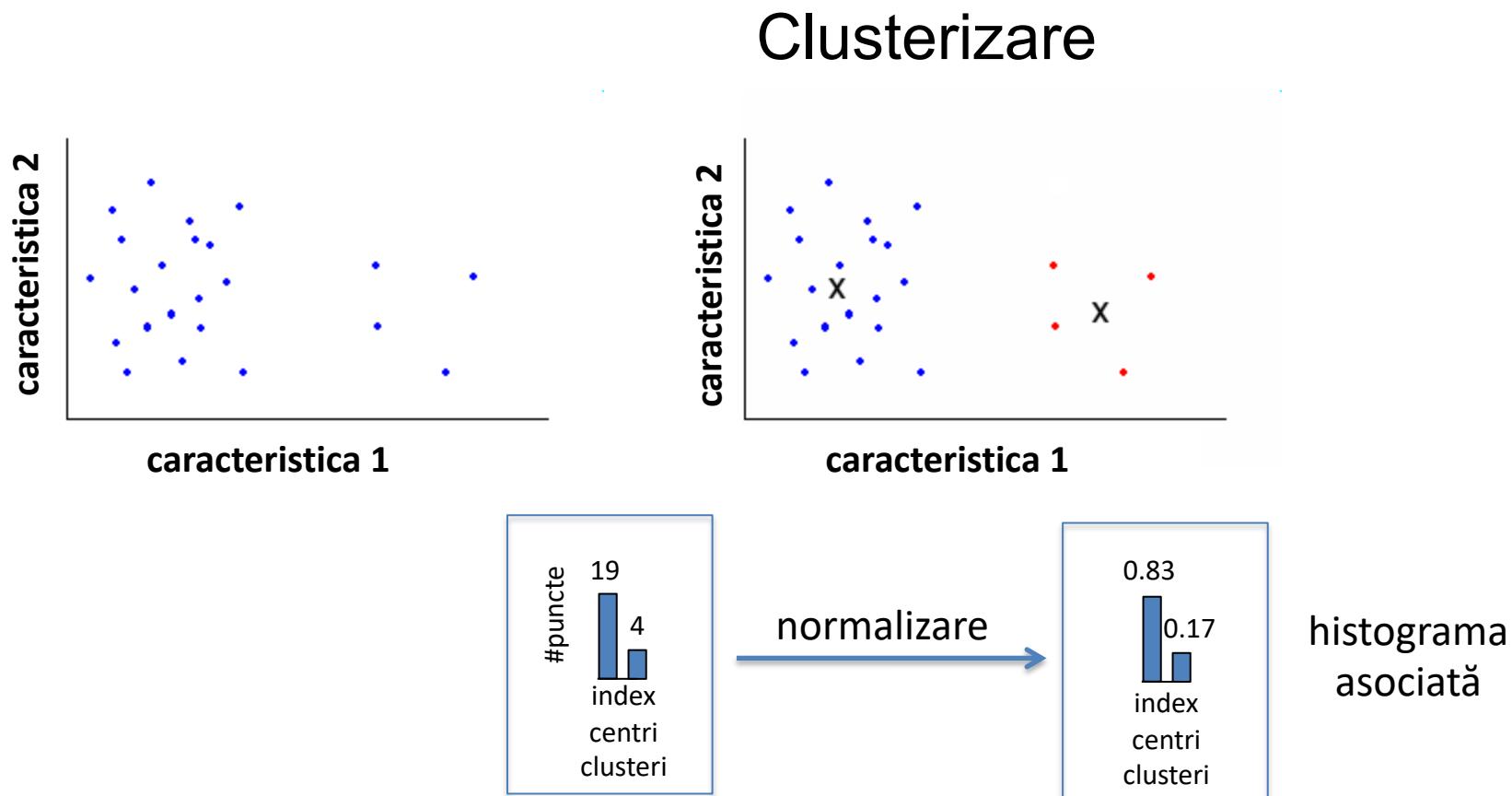
Histograma comună

- 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

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

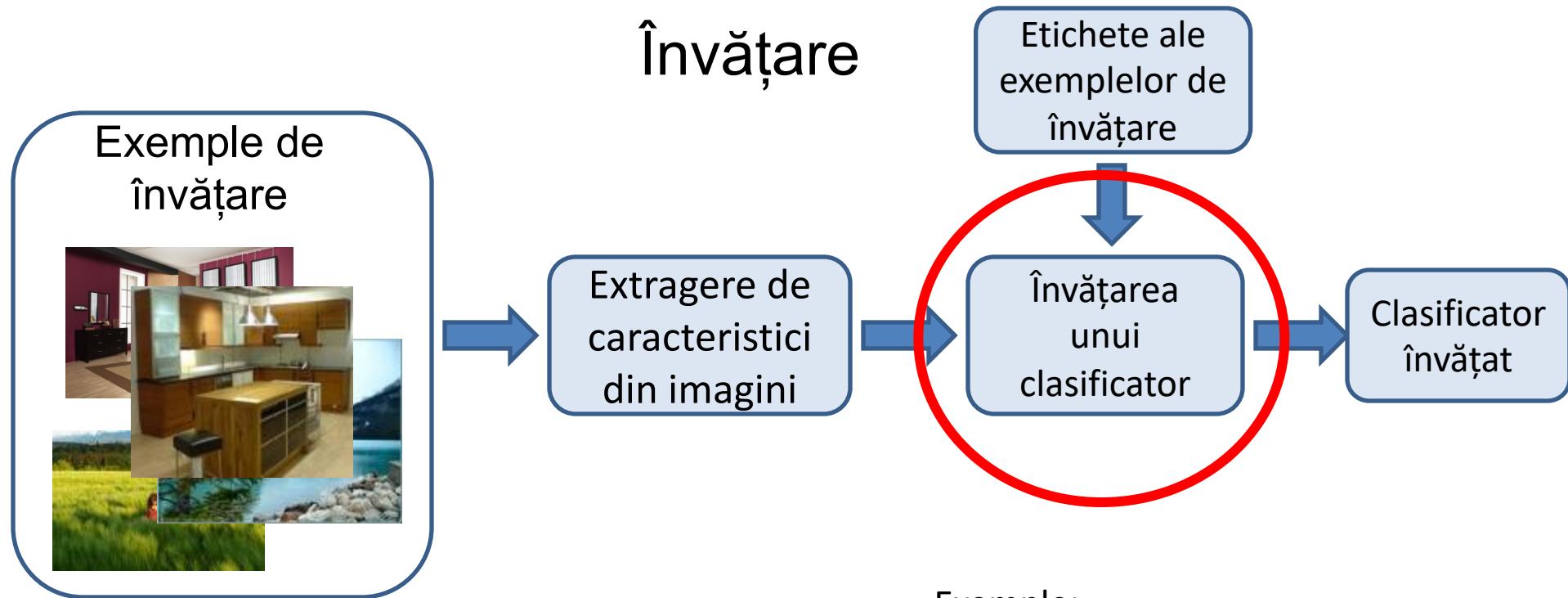
- Distanța dată de intersecția histogramelor

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



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

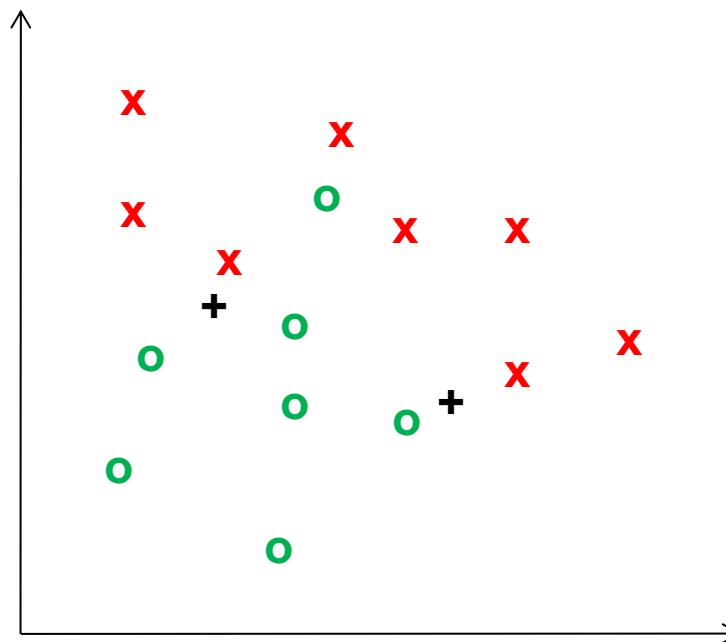
Învățarea unui clasificator



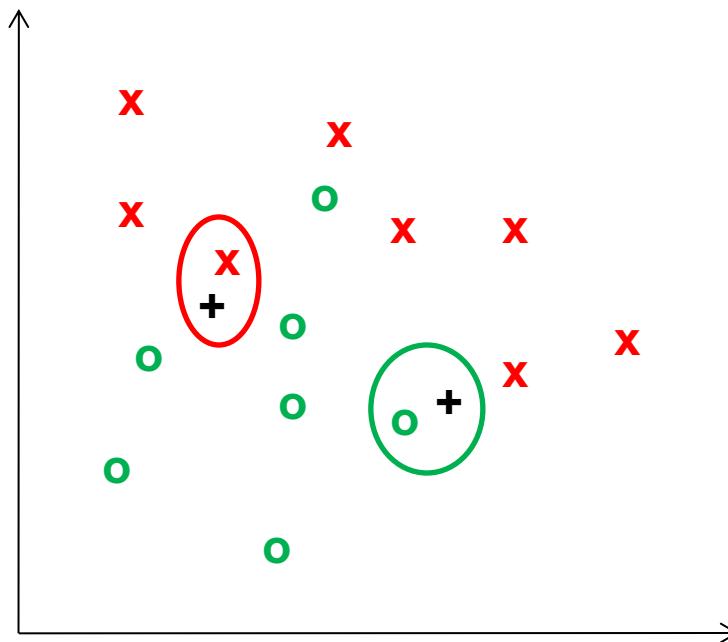
Exemple:

- cei mai apropiati K-vecini
- clasificator liniar

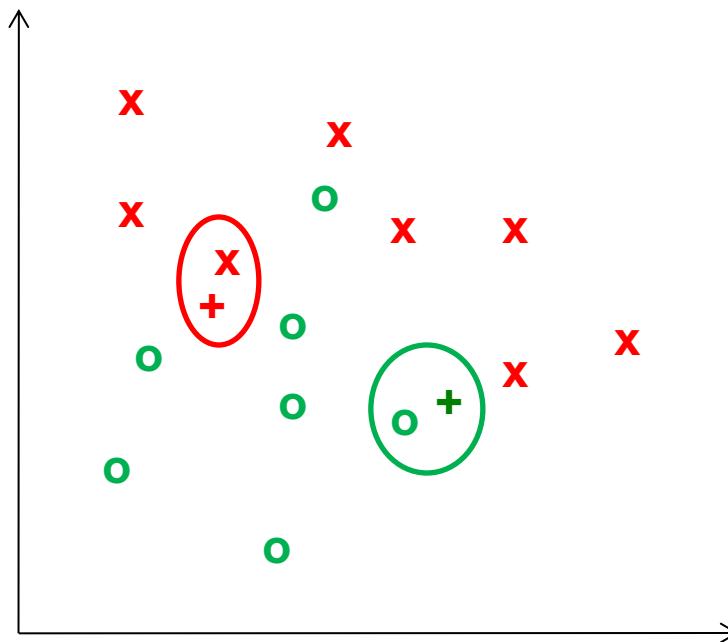
Clasificatorul bazat pe cei apropiati K -vecini



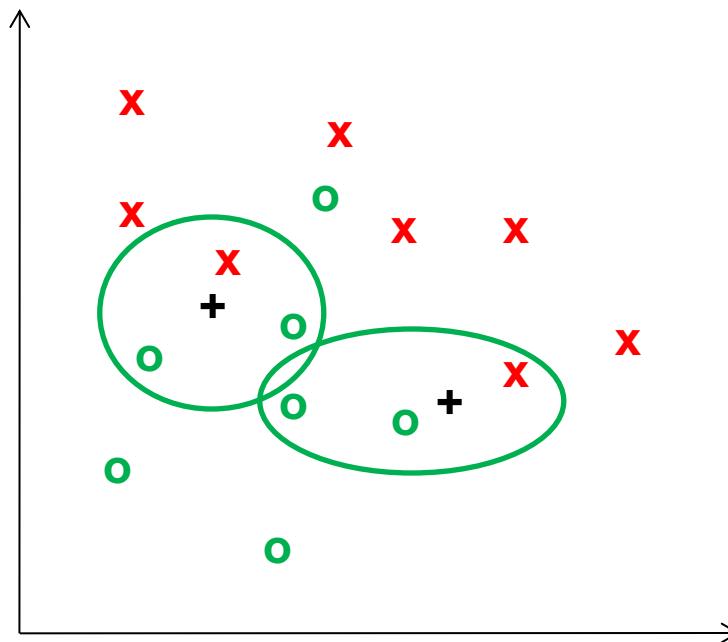
K=1: cel mai apropiat vecini



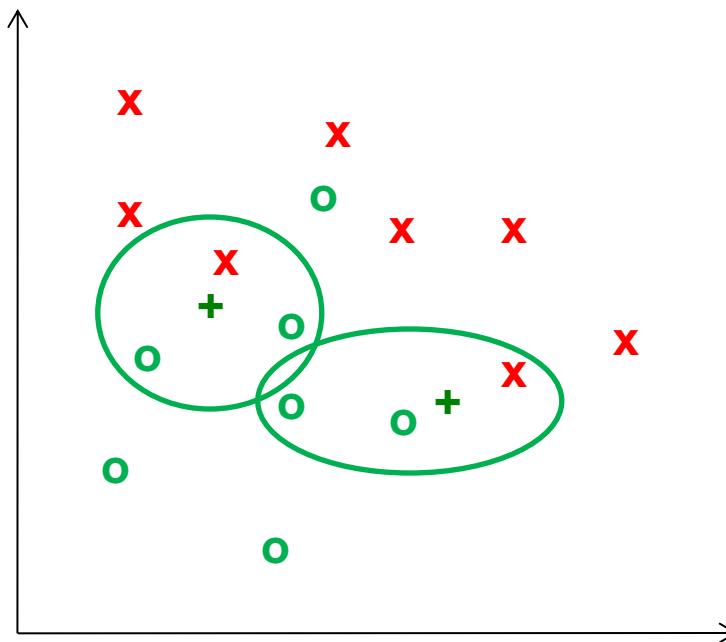
$K=1$: cel mai apropiat vecini



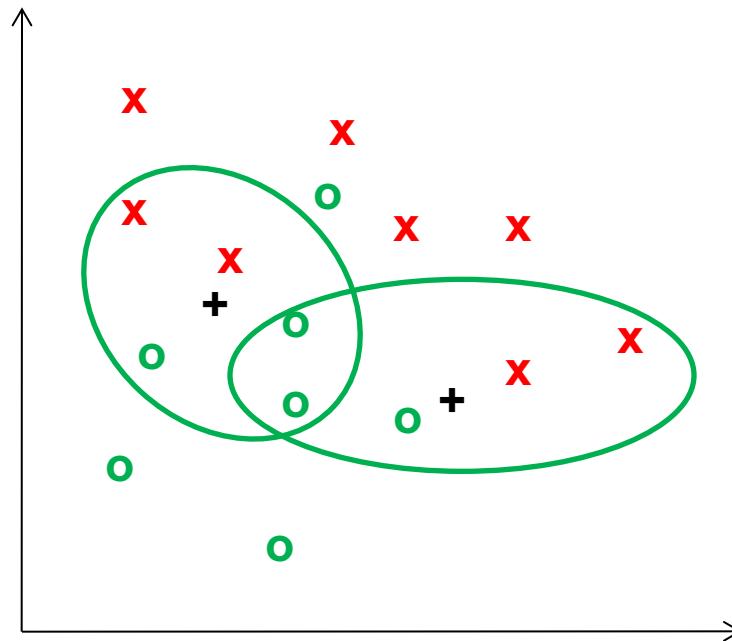
K=3: cei mai apropiati 3 vecini



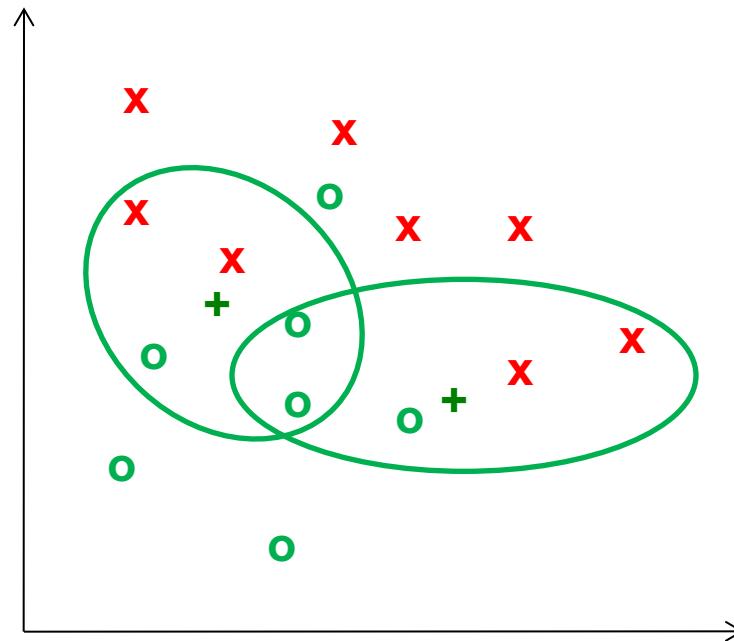
K=3: cei mai apropiati 3 vecini



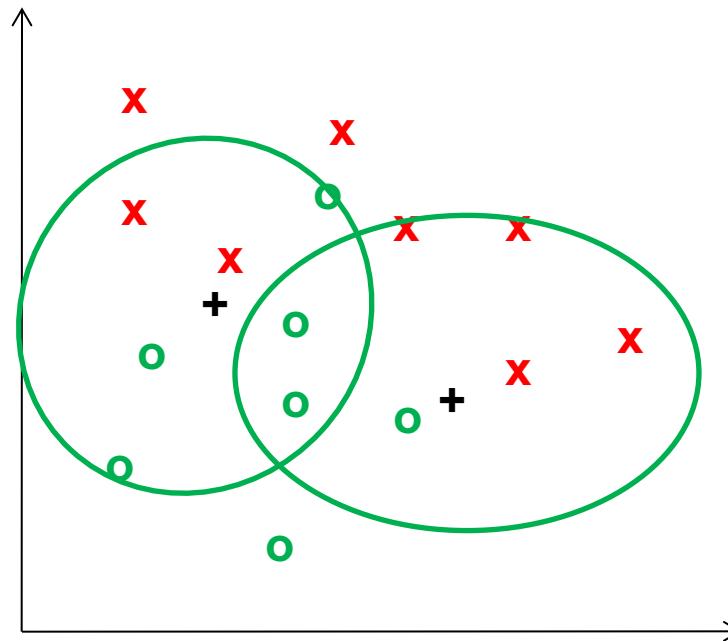
K=5: cei mai apropiati 5 vecini



K=5: cei mai apropiati 5 vecini

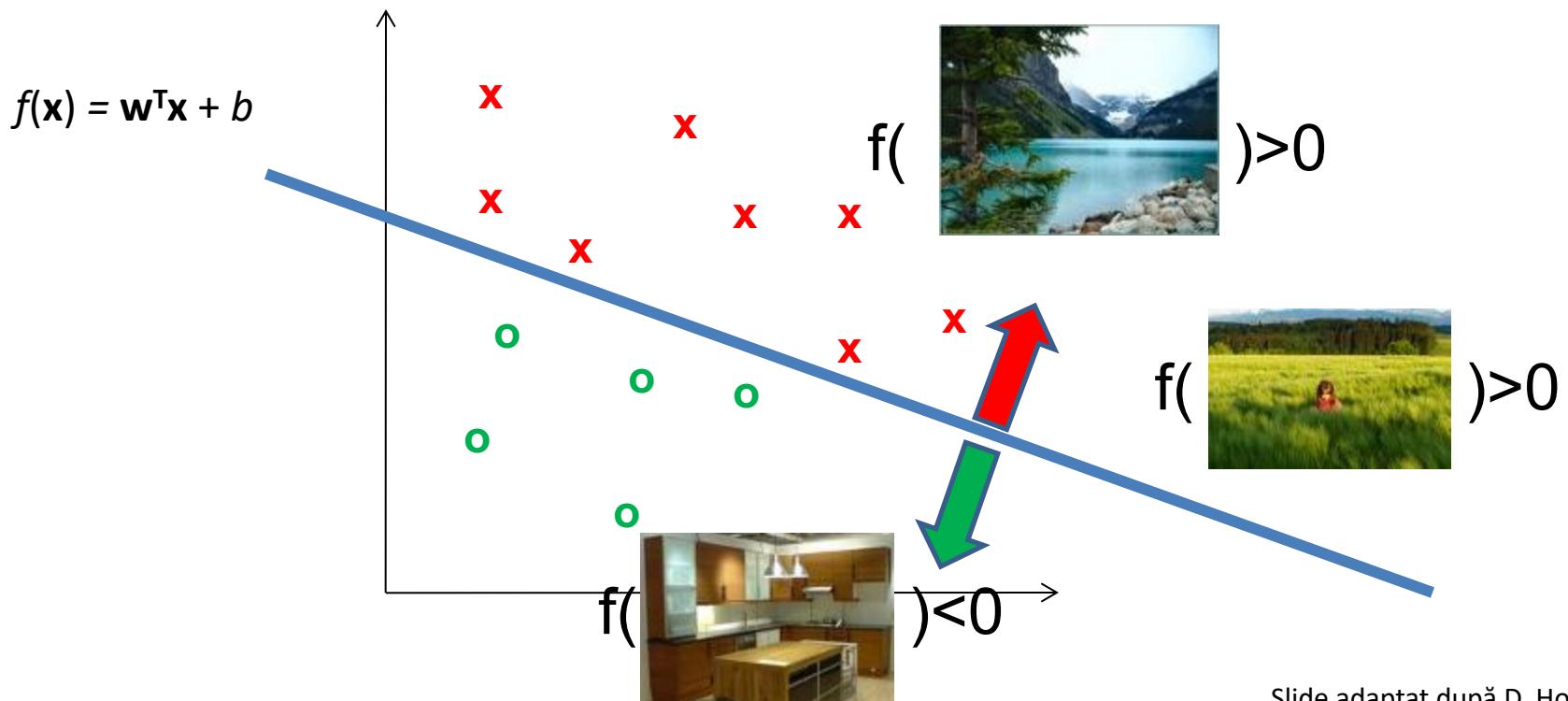


Cei mai apropiati K-vecini



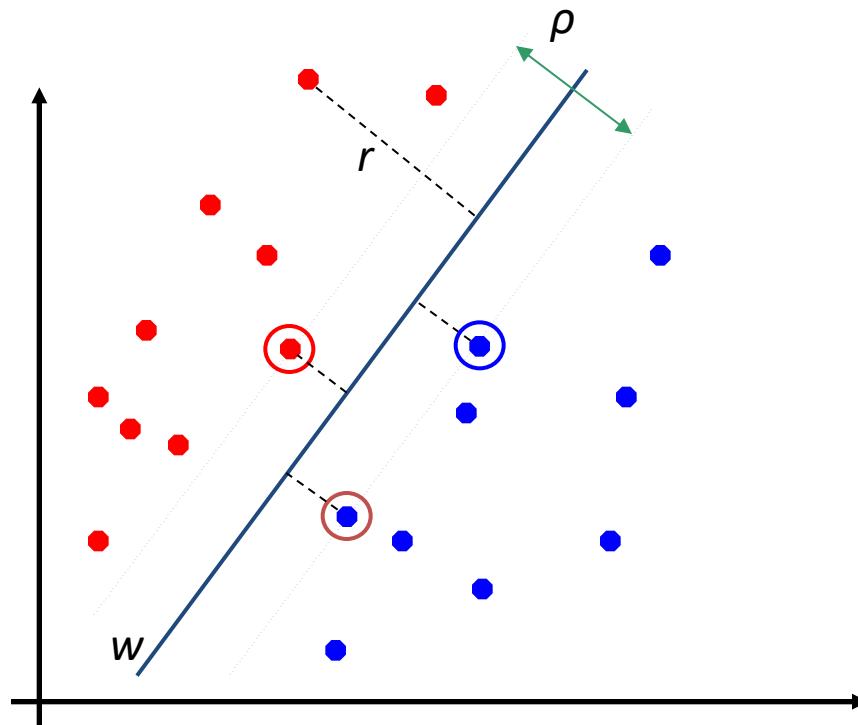
Clasificatorul liniar

Găsește hiperplanul care separă exemplele din clase diferite



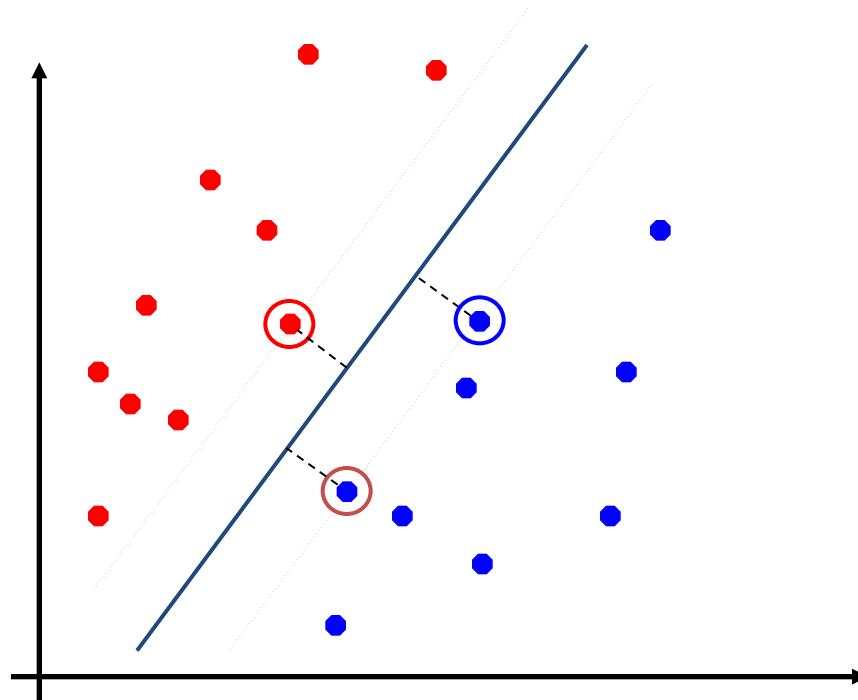
Marginea de clasificare

- 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** ρ 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 ρ e maximă*



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

Recunoașterea claselor de obiecte la nivel generic

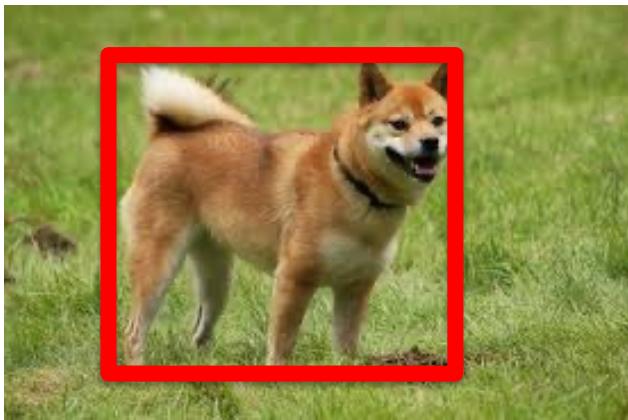
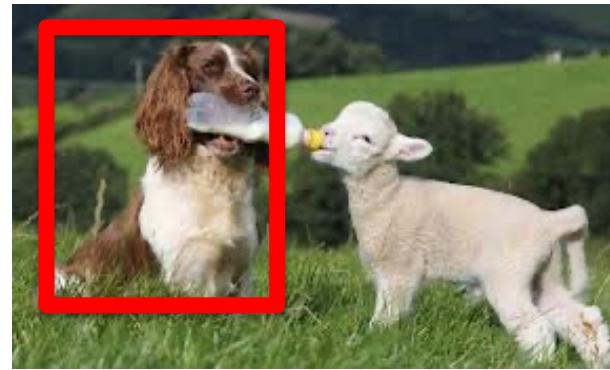
- Mașină



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

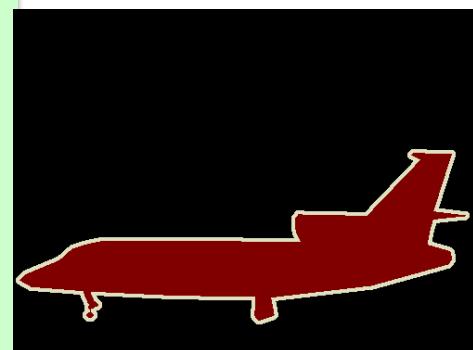
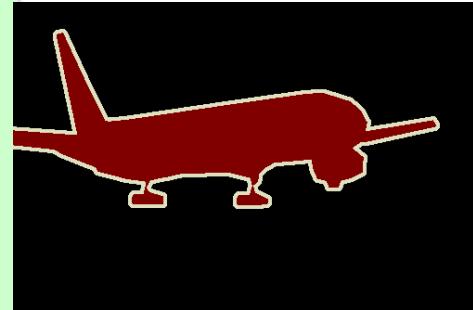
imagini inițiale



localizare - fereastră



localizare - contur



Dificultăți: robustețe



Iluminare



Postura obiectului



Mărime



Mascarea
obiectului



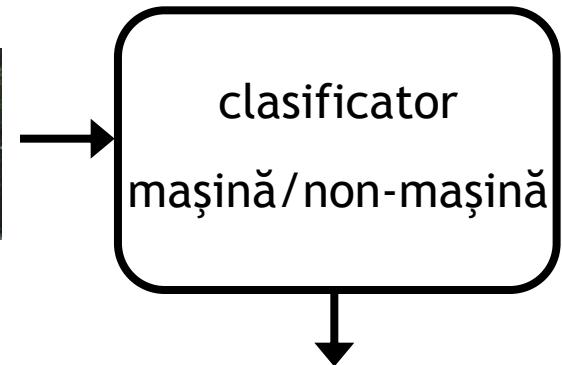
Variabilitate
înfățisare intra-clasă



Poziționarea
camerei

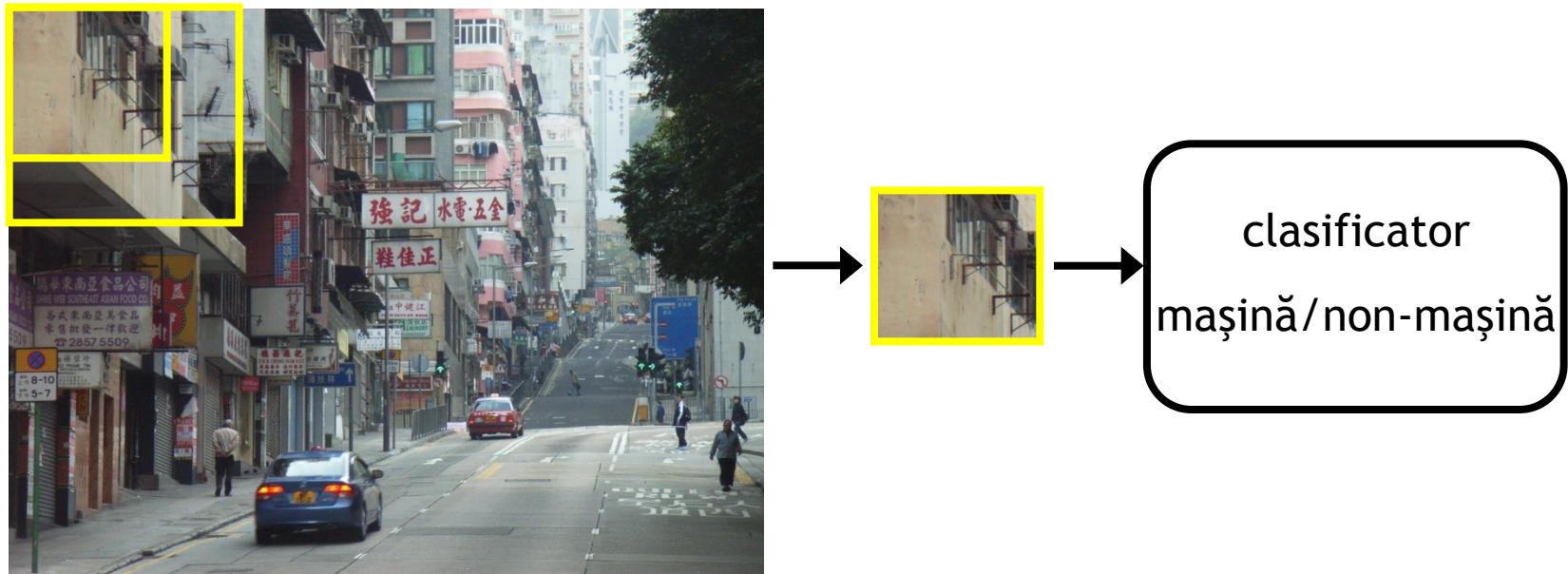
Ideea principală: detectare via clasificare

Componența de bază: un clasificator binar



Ideea principală: detectare via clasificare

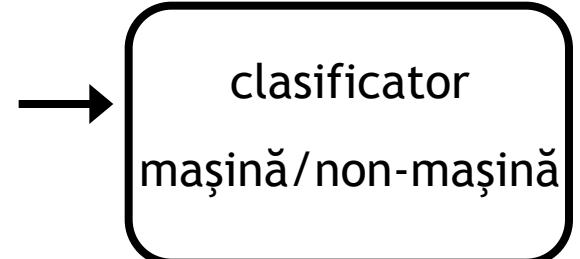
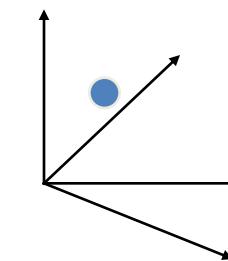
Dacă obiectul este într-o scenă aglomerată, glisează o fereastră căutând obiectul.



Ideea principală: detectare via clasificare

Pași:

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



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?”