

Concepte și Aplicații în Vederea Artificială - Lucrare de laborator, 12 mai 2023 -

Obiectiv

Scopul acestei lucrări de laborator este implementarea unui clasificator care are aplicabilitate în recunoașterea facială precum și implementarea unui sistem automat de extragere a informației vizuale din careuri cu jocul **Avioane**. Datele cu care veți lucra se găsesc în directorul *data* și sunt împărțite pe subdirectoare.

Exercițiul 1 - clasificare binară între membrii familiei Anderson și celelalte personaje - 4 puncte

Pentru primul exercițiu aveți de implementat un clasificator binar care distinge (Figura 1) între imagini din două clase (personaje din familia Anderson și personaje care nu fac parte din familie). În directorul *ex1* se află 250 de imagini cu personaje din serialul de desene animate *Viața cu Louie*. Scopul vostru este să implementați un clasificator binar pentru a distinge între personajele care fac parte din familia Anderson (personajele Andy, Louie, Ora și Tommy) și personaje care nu fac parte din familie dar sunt prezente în serial (vecini,



Figura 1: Exemple de imagini de antrenare: împărțim imaginile în două clase membrii familiei Anderson și personaje care nu fac parte din familie

prieteni etc.). În directorul *ex1* sunt 250 de imagini, 125 de imagini din fiecare clasă. Imaginile din aceeași clasă sunt consecutive: primele 125 de imagini sunt cu membrii familiei Anderson iar următoarele 125 cu personaje ce nu fac parte din familie. Considerăm în cele ce urmează primele 100 de imagini din fiecare clasă (de exemplu, pentru prima clasă, cea cu membrii familiei Anderson, imaginile *0001.jpg* - *0100.jpg*) ca fiind imagini de antrenare și următoarele 25 de imagini (de exemplu pentru clasa cu membrii familiei Anderson, imaginile *0101.jpg* - *0125.jpg*) ca fiind imagini de testare.

ATENȚIE! Folosiți împărțirea datelor în mulțimile de antrenare și testare așa cum este indicat în enunț.

Antrenați-vă un clasificator binar (orice clasificator vreți voi) pe datele de antrenare (200 de imagini din ambele clase) care să distingă între imaginile celor două clase și calculați performanța (acuratețea) lui pe datele de testare (50 de imagini din ambele clase). Explicați în comentarii ce caracteristici folosiți și ce metodă de clasificare binară folosiți.

Punctajul aferent soluției furnizate va fi calculat ca fiind $\text{acuratețe} \times 4$ puncte (o acuratețe de 60% = 2.4 p).

Exercițiul 2 - extragerea automată a informației vizuale pentru jocul Avioane - 5 puncte

Pentru al doilea exercițiu aveți de implementat un sistem automat pentru clasificarea loviturilor unui jucător folosind extragerea informației vizuale din careuri cu jocul **Avioane**.

Jocul **Avioane** este un joc cu 2 jucători, fiecare dintre aceștia începe cu un careu de 10×10 celule pe care poziționează 6 avioane de dimensiuni diferite (3 avioane ce ocupă câte o celulă, 2 avioane ce ocupă 4 celule, și un avion care ocupă 10 celule, avioanele ce ocupă mai mult de o celulă pot avea diferite orientări). Fiecare jucător alege să atace o anumită celulă cu scopul de a doborî avioanele adversarului înainte ca ale sale să fie doborâte. Un avion este doborât atunci când sunt atinse toate celulele sale.

În directorul *ex2* se află 2 directoare, *antrenare* și *testare*. Directoarele *antrenare* și *testare* au aceeași structură, deși datele din directorul *testare* vor fi disponibile după testul de laborator la adresa <https://tinyurl.com/CAVA-12052023-ex2-testare>. Datele constau din imagini cu careuri cu jocul **Avioane** în care sunt plasate avioanele iar fiecare imagine este însoțită de un fișier text ce conține loviturile adversarului. Scopul vostru este să clasificați loviturile adversarului, dacă au atins o țintă sau nu.. Toate imaginile sunt realizate cu același telefon mobil aflat deasupra careului. Este posibil ca luminozitatea scenei să prezinte mici variații de la o imagine la alta. Directorul *antrenare* conține datele de antrenare constând în 10 de imagini. Fișierul cu lovituri corespunzător are același format, extensia *.jpg* fiind înlocuită de extensia *.txt*.

Fișierele cu lovituri conțin următoarele informații

- numărul de lovituri efectuate;
- poziția celulelor care au fost lovite, astfel pentru a specifica o poziție folosim numere

de la 1 la 10 pentru linii iar literele de la A la J pentru coloane, celulele fiind numerotate de sus în jos, de la stânga la dreapta;

Scriveți un program în Python/Jupyter notebook care clasifică loviturile din fișierul cu lovituri și generează fișierele de output adăugând pentru fiecare lovitură 0 sau 1 (0 în situația în care jucătorul nu a lovit un avion și 1 în caz contrar). Distribuția datelor de testare este aceeași ca la antrenare, imaginile fiind obținute în aceleași condiții.

În Figura 2 puteți observa un exemplu de date de intrare și outputul așteptat pentru a patra imagine din setul de antrenare. Unul din jucători are configurația prezentată în careul de joc. Adversarul său alege să atace celulele din fișierul cu lovituri. Scopul vostru este să generați fișierul cu clasificarea loviturilor, 1 pentru cele care lovesc avioanele și 0 în caz contrar.

Pentru acest exercițiu, vom folosi următoarele reguli de notare: vom evalua performanța algoritmului vostru pe o mulțime de 10 de imagini de testare. Pentru fiecare imagine de test algoritmul vostru trebuie să furnizeze un fișier text în care fiecare lovituă este clasificată. Veți primi punctajul numai dacă algoritmul vostru furnizează clasificarea corectă a tuturor loviturilor. Nu există punctaje parțiale. Fiecare configurație corectă valorează 0.5 puncte pentru un total de **5 puncte**;

	A	B	C	D	E	F	G	H	I	J
1										
2		X						X		
3								X		X
4				X				X	X	X
5								X		X
6				X				X		
7			X	X	X					
8								X		
9								X	X	X
10	X						X	X	X	

Figura 2: *Imaginea 004.jpg de antrenare, fișierul cu lovituri și adnotarea corespunzătoare*

Predarea lucrării. Încărcați la adresa <https://tinyurl.com/CAVA-2023-LUCRARE2> o arhivă .zip cu numele vostru și al grupei (spre exemplu 331_ALEXE_Bogdan.zip) cu codul vostru soluție. Includeți în ea eventualele fișiere (modele, etc.) necesare rulării codului.

Observații. Timpul de lucru este de 2 ore. Se acordă 1 punct din oficiu.