

Inteligență Artificială, CTI, anul 3

Test de laborator, 12 ianuarie 2019

În această lucrare veți implementa un sistem de clasificare al imaginilor care conțin scene de *ZI* (clasa 1) sau *NOAPTE* (clasa 0). Directorul ***data*** conține imagini de antrenare și testare pentru cele două clase: *ZI* și *NOAPTE*. Fiind dată o mulțime de antrenare de imagini din cele două clase etichetate (clasa 1 = clasa *ZI*, clasa 0 = clasa *NOAPTE*) vreți să etichetați automat o imagine test.

În directorul ***cod*** veți scrie soluția problemei, completând scriptul ***rezolvaSubiect.m***. Acest script conține codul Matlab pentru citirea și memorarea imaginilor de antrenare din clasa 1 (*ZI*) în structura *IMAGINI*.

În prima parte (cerințele 1-5) veți scrie un clasificator care analizează o imagine la nivel de intensitate grayscale și decide dacă imaginea conține o scenă de *ZI* sau *NOAPTE*. În partea a doua (cerințele 6-9) veți scrie un clasificator care analizează o imagine la nivel de pixeli color și folosește histograme de cuvinte vizuale dintr-un dicționar învățat și decide dacă imaginea conține o scenă de *ZI* sau *NOAPTE*.

Realizați următoarele:

1. (1 punct) Completați codul Matlab pentru a citi și memora în structura *IMAGINI* imaginile de antrenare din clasa 0 (*NOAPTE*).
2. (1 punct) Adăugați în structura *IMAGINI* din codul vostru un câmp cu numele 'medieGrayscale' prin care caracterizați fiecare imagine color prin intensitatea medie a pixelilor săi în reprezentarea grayscale. Practic, pentru fiecare imagine color citită în structura *IMAGINI*, o transformați într-o imagine grayscale (folosiți funcția *rgb2gray*) iar apoi calculați care este media intensităților tuturor pixelilor din această imagine, obținând astfel valoarea câmpului 'medieGrayscale'.
3. (1 punct) Puneți în vectorul *mediiGrayscaleZI* toate valorile câmpurilor 'medieGrayscale' pentru toate imaginile cu eticheta 1 (*ZI*) din structura *IMAGINI*. Calculați media acestui vector obținând astfel *medieZI*. Puneți în vectorul *mediiGrayscaleNOAPTE* toate valorile câmpurilor 'medieGrayscale' pentru toate imaginile cu eticheta 0 (*NOAPTE*) din structura *IMAGINI*. Calculați media acestui vector obținând astfel *medieNOAPTE*. Evident, veți avea *medieNOAPTE* < *medieZI*.
4. (1 punct) Scrieți funcția ***clasificaImagineMedieGrayscale.m*** care primește ca argumente o imagine color *imgColor*, parametri *medieZI* și *medieNOAPTE* și întoarce eticheta 1 (*ZI*) sau eticheta 0 (*NOAPTE*) pe baza următorului raționament: (i) transformă *imgColor* într-o imagine grayscale; (ii) calculează (similar cu punctul 2) media *x* a intensităților tuturor pixelilor din imaginea

grayscale obținută; (iii) dacă x este mai aproape (în sensul distanței Euclidiene) de medieZI decât de medieNOAPTE întoarce eticheta 1, altfel întoarce eticheta 0;

5. (1 punct) Testați-vă clasificatorul scris pe imaginile de testare din directoarele `../data/testare/ZI/` și `../data/testare/NOAPTE/`. Pentru toate imaginile color I din directorul ZI aplicați clasificatorul de la punctul 4 folosindu-vă de valorile medieZI și medieNOAPTE învățate pe mulțimea de antrenare date de structura IMAGINI (punctul 3). Evident, eticheta adevărată a fiecărei imagini I din directorul ZI este 1. Comparați această etichetă cu eticheta decisă de clasificatorul vostru. Calculați și afișați acuratețea clasificatorului vostru (procentul de imagini clasificate corect) pentru imaginile din directorul `../data/testare/ZI/`. Realizați același lucru și pentru imaginile de testare din directorul NOAPTE, având grijă să setați că etichetă adevărată este 0.
6. (1 punct) Extrageți din fiecare imagine de antrenare stocată în structura IMAGINI 100 de pixeli color aleși la întâmplare. Cum aveți 100 de imagini de antrenare din fiecare clasă, aveți în total 200 de imagini de antrenare, deci veți obține o matrice de dimensiune 20000×3 . Clusterizați cei 20000 de pixeli folosind funcția furnizată ***kmeans_iter*** obținând un dicționar C de cuvinte vizuale de dimensiune K . Fiecare linie a matricei C se identifică cu un cuvânt vizual și va reprezenta o culoare. Apelați funcția ***kmeans_iter*** rulând clusterizarea pe 10 iterații și o mărime a dicționarului de $K = 20$ de cuvinte vizuale.
7. (1 punct) Scrieți funcția **`calculeazaHistogramaNormalizataCuvinteVizuale.m`** care pentru o imagine *img* și un dicționar de cuvinte vizuale C calculează histograma normalizată de cuvinte vizuale. Pentru fiecare pixel color din imaginea *img* se identifică cel mai apropiat (în sensul distanței Euclidiene) cuvânt vizual din C și se incrementează frecvența apariției cuvântului vizual în histograma asociată. La sfârșit se normalizează histograma.
8. (1 punct) Obțineți histogramale normalizate de cuvinte vizuale pentru cele 100 de imagini de antrenare din directoarele ZI și NOAPTE folosindu-vă de funcția de la punctul 7. Folosiți dicționarul C obținut la punctul 6.
9. (1 punct) Pentru fiecare imagine de testare din directoarele ZI și NOAPTE obțineți histograma de cuvinte vizuale asociată și realizați clasificarea ei pe baza celui mai apropiat vecin comparând cu cele 200 de histogramme de la punctul 8. Care este acuratețea pe imaginile de testare din cele două directoare?

Timp de lucru 2 ore. Din oficiu 1 punct.

La sfârșit, veți preda pe stick un folder cu numele și grupa voastră conținând numai fișierele Matlab scrise (fără nicio imagine).