

Raport Proiect: Clasificare imaginilor

-Snowboarders vs skiers-

Scopul si descrierea proiectului

Scopul proiectului este utilizarea algoritmului k-Means pentru a obține clasificarea imaginilor in snowboarderi si schiori. Acest lucru poate fi util in dezvoltarea unui sistem automatizat pentru recunoașterea acestora într-un eveniment sportiv, sau, in cazul meu, pentru a învăța si a înțelege cum funcționează algoritmul de clasificare k-Means.

Modul de obținere si de organizare a datelor

Setul de date de antrenare l-am obținut inițial prin utilizarea unui script făcut in python pentru a descărca imagini de pe internet (un image scraper). Astfel am obținut aproximativ 400 de imagini pentru fiecare categorie (schiori si snowboarderi). Acest lucru l-am făcut pentru a scuti munca de a descărca manual fiecare poza de pe internet, însă după utilizarea scriptului, am verificat fiecare imagine si am făcut o triere a lor astfel încât ele sa fie relevante temei alese. Așadar setul de date de antrenare are 175 de poze pentru categoria „schiori” si 123 de date pentru categoria „snowboarderi”. Însă, pentru o acuratețe mai buna, am adăugat la acest set aceleași imagini, transformate (rescalate, blurate si răsturnate). Acest set de date de antrenare este organizat in subfoldere („schiori”, „snowboarderi”), necesare etichetărilor.

De asemenea am si un set de date pentru testarea modelului, având in total 12 imagini (6 pentru schiori si 6 pentru snowboarderi).

Algoritmului utilizat si parametrii acestuia

Algoritmul utilizat este k-Means, fiind unul dintre cei mai cunoscuți algoritmi de învățare nesupervizata care se aplica pe date neetichetate pentru a le împărți în mai multe clusetere (grupuri), fiecare cluster conținând un set de obiecte dintr-o anumita categorie. In cazul meu, numărul de clustere este egal cu 2, reprezentând cele 2 categorii: schiori si snowboarderi. Parametrii utilizați sunt:

- `n_cluster=2`. Acest parametru seteaza numarul de clustere in care vor fi împărțite datele
- `random_state=0`. Acest parametru controlează reproductibilitatea rezultatelor atunci când algoritmul implica aspecte aleatorii (asigura ca algoritmul va genera aceleași rezultate atunci cand este rulat pe aceleași date de intrare).

Biblioteci Python utilizate

- `os`: Pentru manipularea structurii de directoare și fișiere.
- `cv2` (OpenCV): Pentru citirea și prelucrarea imaginilor.
- `numpy`: Pentru manipularea eficientă a datelor în formă de matrice.
- `sklearn.cluster.KMeans`: Pentru implementarea algoritmului K-Means.
- `sklearn.preprocessing.StandardScaler`: Pentru scalarea datelor.
- `matplotlib.pyplot`: Pentru vizualizarea imaginilor și rezultatelor.

Rezultate obținute

Fiecare imagine conține si un text cu rezultatul obținut

Test Image 8 - ==SNOWBOARDER==



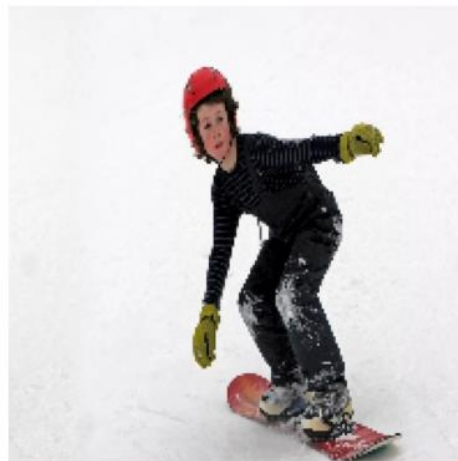
Test Image 6 - ==SNOWBOARDER==



Test Image 1 - ==SCHIOR==



Test Image 10 - ==SNOWBOARDER==



Test Image 11 - ==SNOWBOARDER==



Test Image 3 - ==SCHIOR==



Test Image 9 - ==SNOWBOARDER==



Test Image 4 - ==SCHIOR==



Test Image 5 - ==SCHIOR==



Test Image 7 - ==SCHIOR==



Test Image 12 - ==SCHIOR==



Test Image 2 - ==SCHIOR==



Se poate observa ca din 12 imagini, doar 3 au o etichetare greșită (test image 6, test image 7 si test image 12). Aceste greșeli sunt cauzate in principal din faptul ca cele 2 categorii sunt foarte asemănătoare între ele. Doar poziția omului, ar face diferența, însă de multe aceasta este asemănătoare in ambele cazuri. Placa sau schiurile de asemenea ar putea face diferența între cele 2 categorii, însă nu

Întotdeauna acestea sunt vizibile, din cauza zăpezii, care este prezenta în ambele cazuri, astfel fiind chiar greu și pentru un om să își dea seama.