**Animals Classification**

Andrei Antonia-Ștefania

333AA

Cuprins

1. Introducere : Motivație, descrierea temei, tip de algoritm și algoritm ales
2. Metodologie: Descriere implementare și particularități algoritm
3. Rezultatele obținute la antrenare, validare și testare.Matricea de confuzie și indicatorii de performanță
4. Concluzii

1.Introducere

Aplicația are ca scop clasificarea imaginilor a 20 de categorii de animale folosind algoritmul K-Means Clustering. Aceasta este o metodă nesupervizată de învățare automată, care grupează datele în clustere pe baza similarităților.

Clasificarea imaginilor este un domeniu important în inteligența artificială și viziunea computerizată, fiind aplicată în diverse industrii, precum medicină, monitorizare, recunoaștere facială sau recunoașterea speciilor animale. În acest proiect, motivația principală este dezvoltarea unui sistem care să clasifice automat imagini conținând 20 de categorii de animale. Această problemă are o complexitate ridicată, deoarece diferitele animale pot avea trăsături similare, iar iluminarea, unghiurile și calitatea imaginilor pot influența rezultatele.

Sistemul procesează imaginile din seturile de antrenare, validare și testare, extrage caracteristici sub formă de vectori și le utilizează pentru a identifica clustere (grupuri) reprezentative pentru fiecare categorie. Algoritmul asociază fiecare imagine celui mai apropiat centru al clusterului, identificat prin minimizarea distanței dintre caracteristicile imaginii și centrul clusterului.Ulterior, clusterele sunt asociate cu etichete reale pentru a permite evaluarea performanței algoritmului.

Am optat pentru algoritmul KMeans, care este unul dintre cei mai cunoscuți algoritmi de clustering nesupervizat. KMeans funcționează prin:

* Împărțirea datelor în k clustere (unde k este numărul de categorii)
* Calcularea centroidului fiecărui cluster
* Atribuirea fiecărui punct (imagine) clusterului cu cel mai apropiat centroid
* Recalcularea centrelor până la convergență.

Acest algoritm este eficient în probleme de clustering, dar are limitări în fața datelor complexe sau cu dimensiuni ridicate.

Principiul de funcționare al K-Means:

* Inițializare: Alegerea aleatorie a K centre (numărul de clustere este ales în funcție de numărul categoriilor).
* Atribuirea punctelor: Fiecare punct (imagine) este asignat clusterului al cărui centru este cel mai apropiat, calculat pe baza unei metrici (de obicei distanța Euclidiană).
* Actualizarea centrelor: Centrele clusterelor sunt recalculare ca medie a tuturor punctelor atribuite acelui cluster.
* Iterare: Procesul se repetă până când centrele clusterelor nu se mai schimbă semnificativ sau s-a atins un număr maxim de iterații.

Avantajele utilizării K-Means Clustering:

* Este un algoritm simplu, eficient și rapid pentru seturi de date mari.
* Este util în cazurile în care datele nu sunt etichetate.
* Oferă o bună grupare a datelor pe baza caracteristicilor comune.

Dezavantajele algoritmului:

* Necesită specificarea numărului de clustere (K) înainte de antrenare.
* Performanța poate fi afectată de inițializarea centrelor clusterelor.
* Nu poate identifica clustere cu forme neregulate.

În acest proiect, numărul de clustere (K) este egal cu numărul de categorii de animale din dataset, iar rezultatele obținute sunt comparate cu etichetele reale pentru a evalua eficiența modelului.

2. Metodologie

Proiectul utilizează algoritmul K-Means pentru clasificarea imaginilor dintr-un dataset cu 20 de categorii de animale. Scopul este de a implementa și analiza performanța acestui algoritm de învățare nesupervizată aplicat pe imagini redimensionate și reprezentate prin vectori de caracteristici.Imaginile sunt organizate într-un dataset ce conține trei subfoldere principale:

* Train (antrenare) - folosit pentru învățarea modelului.
* Validation (validare) - utilizat pentru a evalua performanța modelului pe date nevăzute în timpul antrenării.
* Test (testare) - destinat evaluării finale a modelului pentru a estima performanța pe date complet noi.

Librăriile folosite:

* Os - utilizată pentru operațiuni legate de sistemul de fișiere (navigare, gestionare directoare, obținerea listelor de fișiere).
* cv2 (OpenCV) - bibliotecă utilizată pentru procesarea imaginilor.
* numpy (NumPy) - utilizată pentru operații matematice și lucrul cu tablouri multidimensionale (array-uri).
* sklearn.cluster.Kmeans- algoritmul de clustering K-Means din Scikit-learn.
* sklearn.metrics - metrici pentru evaluarea performanței modelului.
* confusion\_matrix: crearea matricei de confuzie.
* classification\_report: rezumatul metricilor de performanță (precizie, recall, F1-score).
* accuracy\_score: Calcularea acurateței generale.
* sklearn.preprocessing.LabelEncoder - utilizată pentru codificarea etichetelor de text (nume de clase) în valori numerice.
* Seaborn - utilizată pentru vizualizări avansate, cum ar fi matricea de confuzie.
* matplotlib.pyplot - utilizată pentru vizualizarea datelor.

Aceste librării combină procesarea imaginilor (OpenCV), analiza datelor (NumPy, Matplotlib, Seaborn) și algoritmi de învățare automată (Scikit-learn).

Dataset-ul a fost descărcat de pe site-ul : <https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals/data> , dar inițial avea 90 de categorii de animale și am redus la 20 de categorii.

Dataset:

Alcătuit din 20 de subfoldere, denumite: bear,butterfly,cat,cow,deer,dog,dolphin,elephant,flamingo,fox,horse,lion,panda, parrot,penguin,pig,sheep,tiger,turtle,zebra. Acestea reprezintă categoriile de animale.

Descrierea implementării

* Preprocesarea imaginilor:
* Imaginile din dataset sunt încărcate și convertite în vectori de caracteristici prin aplatizarea pixelilor. Această metodă transformă fiecare imagine color într-un vector unidimensional de lungime 128×128×3=49,152, unde 128x128 este dimensiunea redimensionată a imaginilor, iar 3 reprezintă cele trei canale RGB.
* Etichetele imaginilor sunt extrase din numele folderelor, fiecare folder reprezentând o categorie de animale.
* Algoritmul K-Means:
* Algoritmul K-Means este aplicat pentru împărțirea datelor în 20 de clustere, corespunzătoare celor 20 de categorii.
* Fiecare cluster este reprezentat de un centroid, calculat ca media punctelor din cluster. Atribuirea unui punct (imagine) la un cluster este realizată prin minimizarea distanței Euclidiene dintre punct și centroid.
* Centroidele sunt inițializate aleator, iar algoritmul iterează până la convergență, ajustând pozițiile centroidelor și atribuțiile punctelor.
* Maparea clusterelor la etichetele reale:
* După antrenare, fiecare cluster este mapat la eticheta reală predominantă dintre imaginile asociate clusterului respectiv, utilizând distribuția etichetelor reale din fiecare cluster.
* Evaluarea performanței:

Performanța este măsurată pe seturile de antrenare, validare și testare, utilizând următorii indicatori:

* Matricea de confuzie: pentru a evalua numărul de clasificări corecte și greșite.
* Acuratețea globală: proporția totală a imaginilor clasificate corect.
* Indicatorii de performanță: precizie, recall și scor F1 pentru fiecare categorie.

Particularitățile algoritmului

* Reducerea dimensionalității: Prin aplatizarea imaginilor, caracteristicile spațiale sunt pierdute. Acesta este un compromis pentru a utiliza un algoritm simplu de clustering, cum este K-Means.
* Performanță dependentă de inițializare: K-Means este sensibil la poziționarea inițială a centroidelor, ceea ce poate afecta convergența.
* Etichetarea post-clustering: Deoarece K-Means este un algoritm nesupervizat, etichetele finale sunt atribuite pe baza etichetelor reale predominant asociate fiecărui cluster.

3. Rezultatele obținute

Rezultatele obținute la antrenare:

A screen shot of a chart

Description automatically generated

Precizia:

* Ex.: Pentru "dolphin", precizia de 78% indică faptul că, dintre toate instanțele prezise ca "dolphin", 78% au fost corecte.
* Clase precum "dog", "fox", "pig" și "parrot" au precizia 0, ceea ce înseamnă că modelul nu a făcut nicio predicție corectă pentru aceste clase.

Recall:

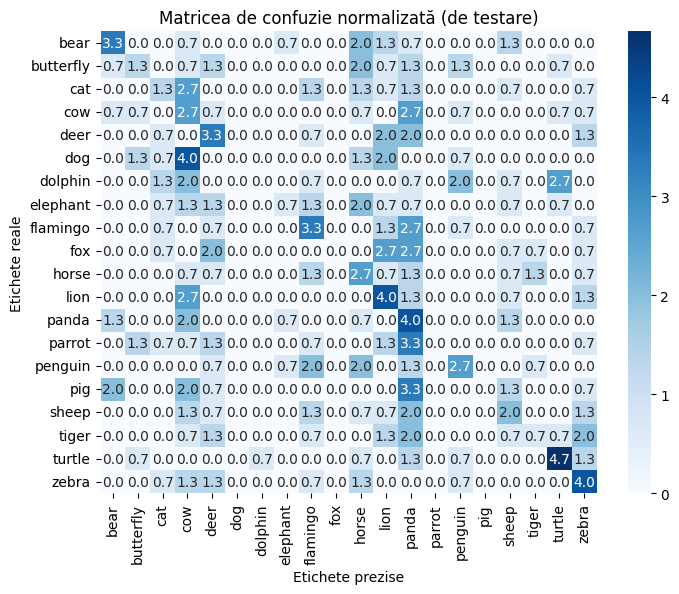
* Ex.: Pentru "deer", recall-ul de 33% arată că modelul a identificat corect 33% din toate exemplele reale de "deer".
* Clase precum "dog", "fox", "pig" și "parrot" au recall 0, ceea ce înseamnă că modelul nu a reușit să detecteze niciun exemplu din aceste clase.

F1-Score:

* Clase precum "turtle" și "penguin" au F1-scores mai bune (0.43 și 0.33), ceea ce sugerează un echilibru relativ între precizie și recall, dar alte clase, cum ar fi "dog", au F1-score 0, ceea ce indică lipsa completă a performanței.
* Rezultatele obținute la validare:  
    
  A screen shot of a data

  Description automatically generated
* Pentru clasa dolphin, precision-ul este 0.83: dintre toate exemplele prezise ca "dolphin," 83% sunt corecte.
* Pentru clasa deer, recall-ul este 0.40: modelul a găsit 40% dintre exemplele reale din clasa "deer."
* Pentru clasa penguin, F1-score-ul este 0.36, sugerând că modelul clasifică această clasă relativ mai bine decât altele.

Rezultatele obținute la testare:



* Precision (precizie): Procentul de predicții corecte pentru o anumită clasă, raportat la totalul predicțiilor făcute pentru acea clasă.

Precision = TP/ (TP + FP)

Exemplu pentru bear: Din totalul predicțiilor făcute ca fiind "bear", doar 42% sunt corecte.

* Recall (rapel/sensibilitate): Procentul de exemple corecte ale unei clase care au fost identificate corect.

Recall = TP/(TP+FN)

Exemplu pentru bear: Modelul identifică corect doar 33% dintre urșii prezenți în datele de test.

* F1-Score: Media armonică între precizie și rapel, echilibrând situațiile în care una este foarte scăzută.

F1 = 2 \* (Precision \* Recall)/ (Precision + Recall)

Exemplu pentru bear: F1-Score este 0.37, ceea ce reflectă o performanță generală slabă (ambele valori sunt mici).  
  
Rezultate :   
  
A bear walking on the ground

Description automatically generatedA butterfly on a flower

Description automatically generated  
  
A close up of a cat

Description automatically generatedA cow standing in a field

Description automatically generated  
  
A deer with antlers standing in a field

Description automatically generated

1. Concluzii

* Algoritmul K-Means a reușit să grupeze imaginile din setul de antrenare în clustere, iar după asocierea clusterelor la etichetele reale, am obținut o corespondență rezonabilă între clustere și categoriile de animale. Acest lucru arată că algoritmul a identificat caracteristici comune ale imaginilor din aceeași categorie.
* Setul de date descărcat a avut un dezechilibru semnificativ în numărul de imagini din fiecare categorie. Acest fapt a generat multe greșeli în matricea de confuzie. Pentru a diminua aceste erori, am redus numărul de imagini din categoriile suprapopulate și am păstrat un număr aproximativ egal de imagini pentru fiecare clasă de animale.
* Performanța modelului pe setul de antrenare a fost rezonabilă, având un nivel acceptabil de clasificări corecte pentru cele mai multe categorii. Aceasta indică faptul că modelul a reușit să învețe caracteristicile principale ale fiecărei clase din setul de antrenare.
* Pe setul de validare, performanța modelului a scăzut comparativ cu setul de antrenare. Din matricea de confuzie, se observă confuzii semnificative între clasele cu trăsături similare, cum ar fi cele între "dog" și "wolf". Aceste confuzii indică o capacitate limitată a modelului de a distinge clasele foarte apropiate ca aspect.
* Rezultatele obținute pe setul de testare confirmă concluziile setului de validare. Modelul întâmpină dificultăți în clasificarea corectă a anumitor clase, în special a celor care au imagini cu fundaluri complexe sau cu trăsături vizuale asemănătoare.
* Examinând imaginile din setul de testare în care modelul a greșit, am observat că cele mai multe erori au fost făcute pentru imagini cu fundaluri detaliate, cu lumini diferite sau pentru animale care seamănă între ele ca formă sau culoare (de exemplu, "tiger" și "leopard").
* Normalizarea matricei de confuzie a evidențiat mai clar distribuția erorilor între clase. Unele clase, precum "zebra" și "turtle", au fost mai ușor de identificat de model, în timp ce clase precum "bear" sau "dolphin" au avut o rată mai mare de confuzii.

Bibliografie

* [Kaggle](https://www.kaggle.com/) - Platformă folosită pentru descărcarea seturilor de date cu imagini de animale.
* Documentație Python pentru manipularea datelor și vizualizare: <https://docs.python.org/>
* Prelucrarea datelor și augmentarea imaginilor
* Albumentations Library: <https://albumentations.ai/>
* PIL (Python Imaging Library): <https://pillow.readthedocs.io/>
* <https://www.geeksforgeeks.org/k-means-clustering-introduction/>
* <https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/>
* <https://www.w3schools.com/python/python_ml_k-means.asp>
* <https://www.geeksforgeeks.org/confusion-matrix-machine-learning/>
* <https://www.statology.org/sklearn-classification-report/>
* <https://www.youtube.com/watch?v=EItlUEPCIzM&t=8s>