Etapa 1

Extragerea de atribute (Feature Extraction)

Am implementat un flux de extragere a atributelor folosind două metode principale: descriptori ORB şi reducerea dimensionalității cu PCA.

1. Preprocesarea imaginilor

Am redimensionat imaginile la o dimensiune standard de **32x32 pixeli** pentru a uniformiza setul de date, ceea ce reduce complexitatea calculelor și asigură consistență între imagini.

2. Extragerea descriptorilor ORB

Am utilizat ORB pentru a extrage caracteristici semnificative din imagini, cum ar fi colțuri și modele locale.

- **Parametrii aleşi**:
    - `max_keypoints=300:` permite detectarea a până la 300 de puncte de interes pentru fiecare imagine.
    - `edge_threshold=6` pentru controlul sensibilitatii la marginile imaginii
- **Justificare**: ORB este rapid şi eficient pentru imagini în tonuri de gri, ceea ce îl face potrivit pentru seturi mari de date. Este rezistent la modificări de scalare sau rotație.

3. Clustering cu KMeans (Bag of Words)

Descriptorii extraşi au fost grupați în **500 clustere** utilizând KMeans, pentru a construi un vocabular vizual.

- Descriptorii din toate imaginile au fost concatenați într-un set comun, iar KMeans a fost aplicat pentru a învăța centrele clusterelor.
- Fiecare imagine a fost reprezentată ca o histogramă bazată pe frecvența clusterelor corespunzătoare descriptorilor săi.
- **Justificare**: Bag of Words oferă o reprezentare compactă și numerică a imaginilor, reducând complexitatea analizei ulterioare.

4. Reducerea dimensionalității cu PCA

Am aplicat PCA pentru a reduce dimensiunea histogramelor la **20 de componente principale**.

După generarea histogramei descriptorilor, dimensiunea acestora poate fi foarte mare. PCA este folosit pentru a reduce numărul de dimensiuni la un număr mai mic de componente, păstrând în același timp cât mai mult din variabilitatea originală a datelor.

- Am standardizat datele înainte de PCA pentru a avea o medie de 0 şi o varianță unitară.

- **Justificare**: PCA reduce redundanța dintre caracteristici, îmbunătățind eficiența modelelor și explicând ~95% din variabilitatea datelor cu mai puține atribute.

5. Selecția atributelor

Am utilizat metoda **VarianceThreshold** pentru a elimina caracteristicile cu varianță scăzută (sub 0.01), care nu sunt utile pentru discriminarea între clase. Aceasta păstrează doar caracteristicile relevante pentru clasificare.

- Selecția atributelor ajută la reducerea zgomotului și a timpului de procesare, îmbunătățind performanța modelului final.

Rezultate și observații

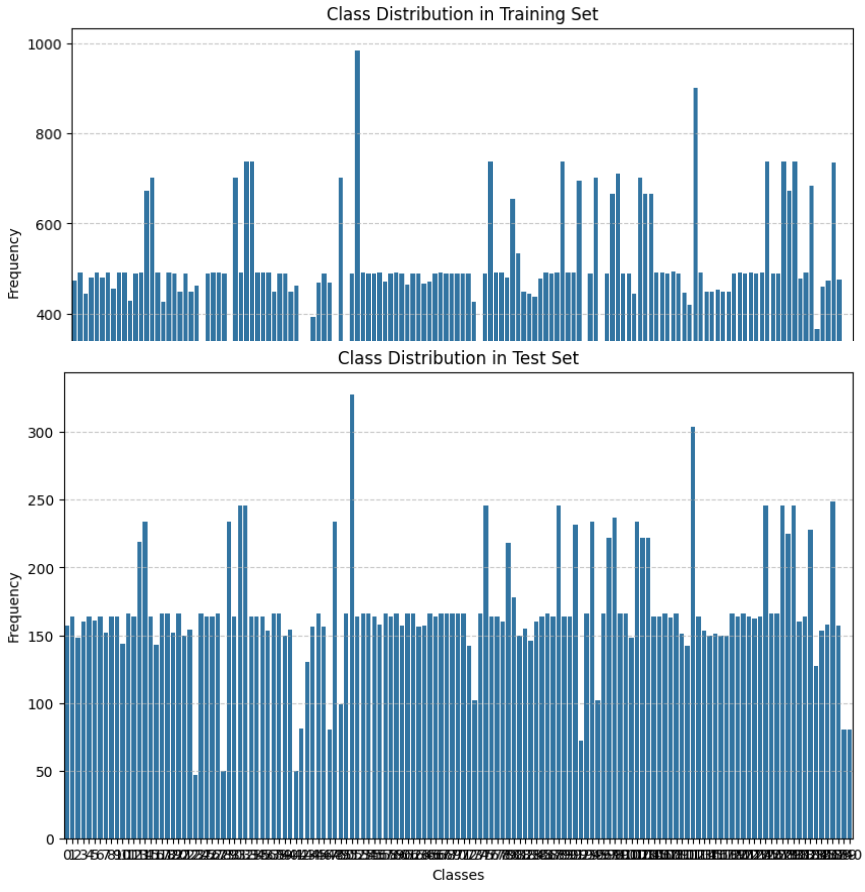
- **Dimensiuni finale**: După fluxul de preprocesare, dimensiunea setului de antrenament a fost redusă semnificativ la *(n_samples, 10)*, unde 10 este numărul de caracteristici finale.
- Metodele alese oferă o bună echilibrare între precizie și eficiență, reprezentând bine imaginile cu un model numeric compact și optimizat.

Acest flux poate fi extins sau ajustat în funcție de performanțele modelelor de clasificare ulterioare.

Vizualizarea atributelor extrase

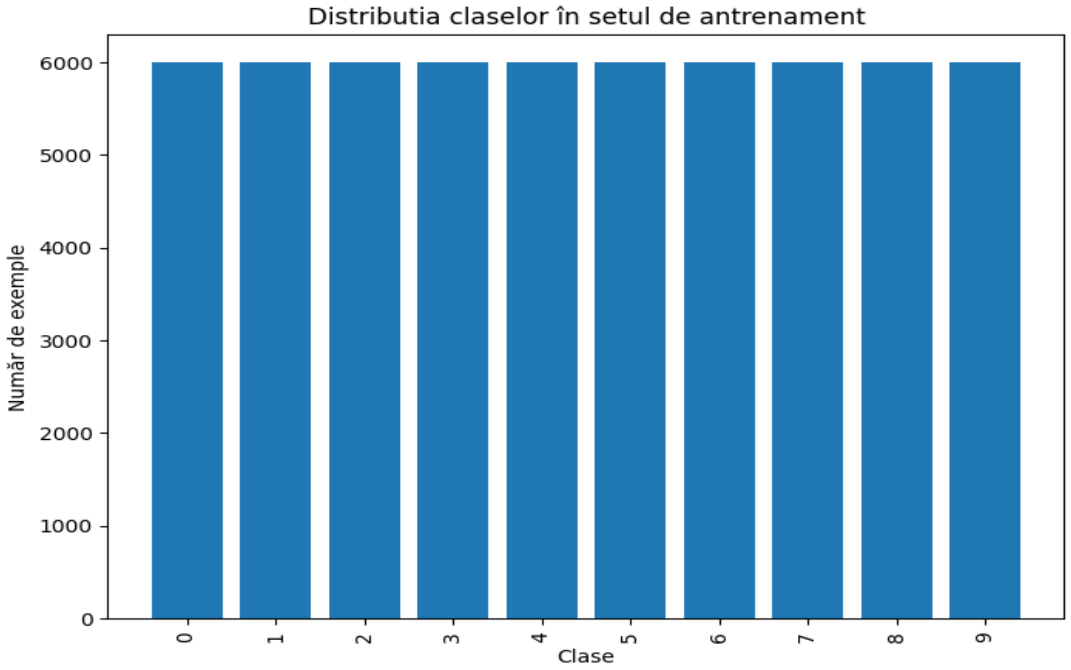
Distributia claselor

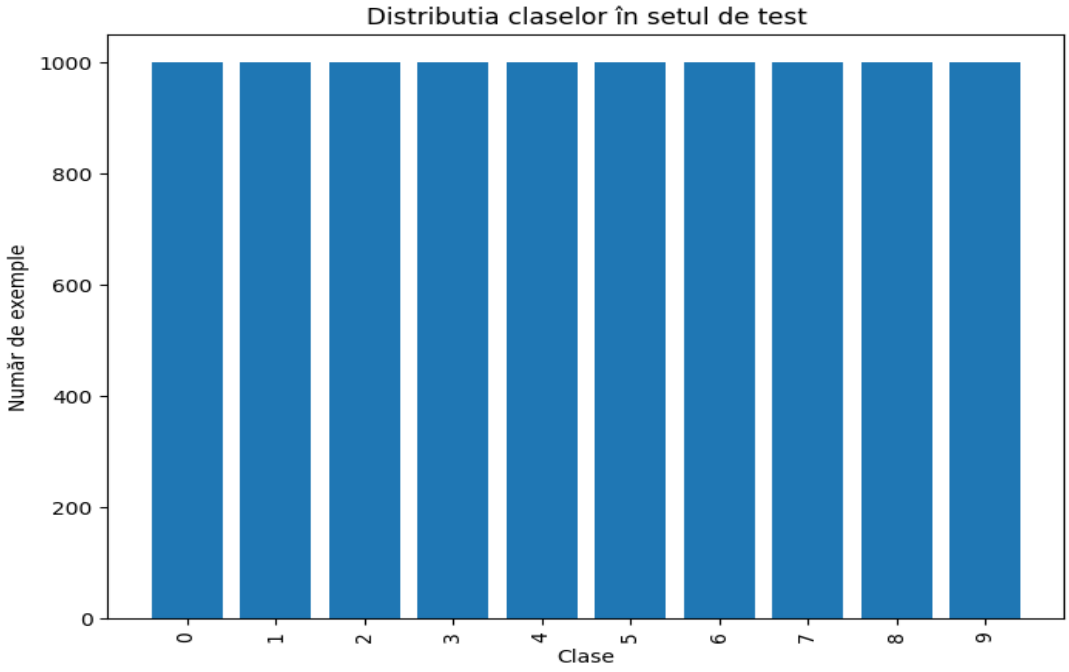
Fruits
Pentru ambele seturi de date se observa un dezechilibru intre clase, cu unele sub-reprezentate si cateva supra-reprezentate , dar totusi se observa ca pentru antrenament "se converge" pe o mede de 500 de poze per clasa, iar pentru testing o medie de 150.



Fashion

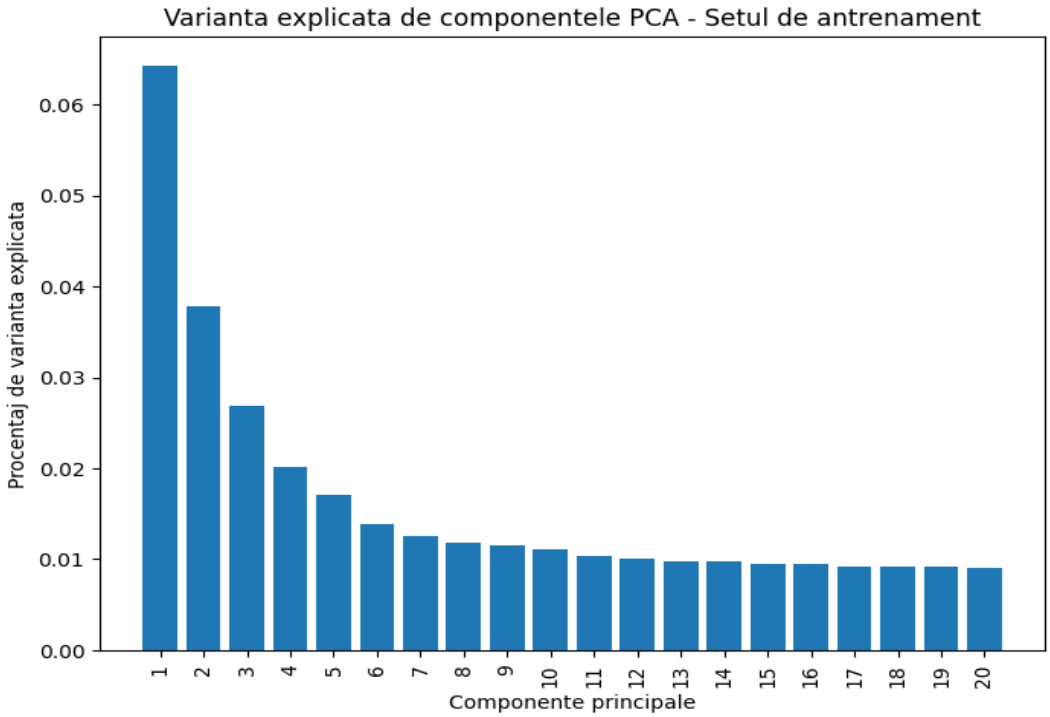
Se poate observa ca toate clasele sunt echilibrate atat pentru setul de testare cat si pentru cel de antrenament.



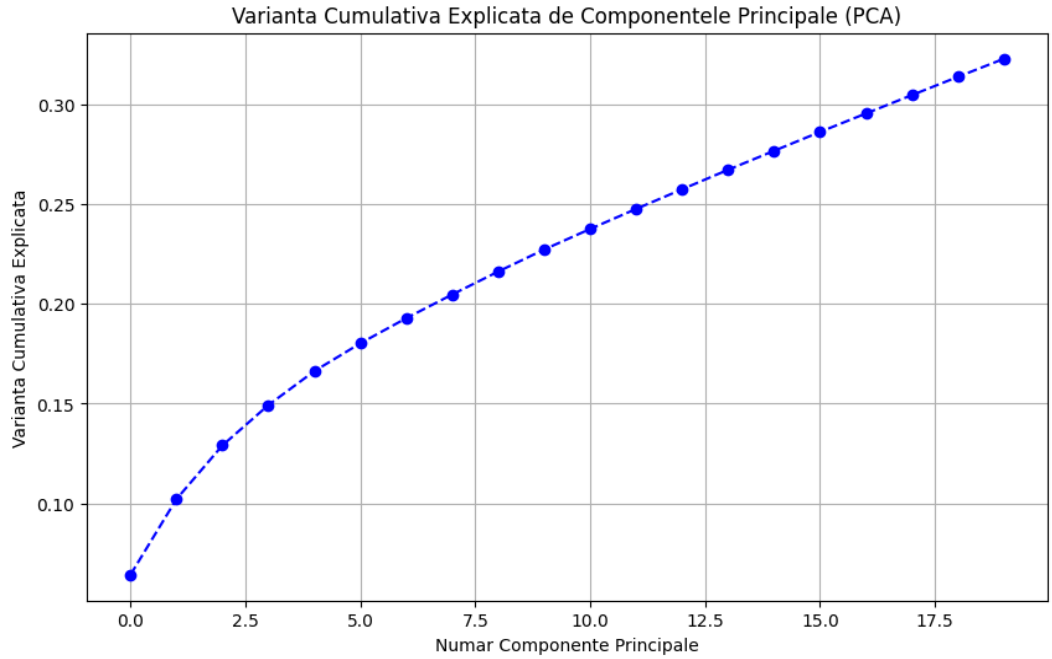


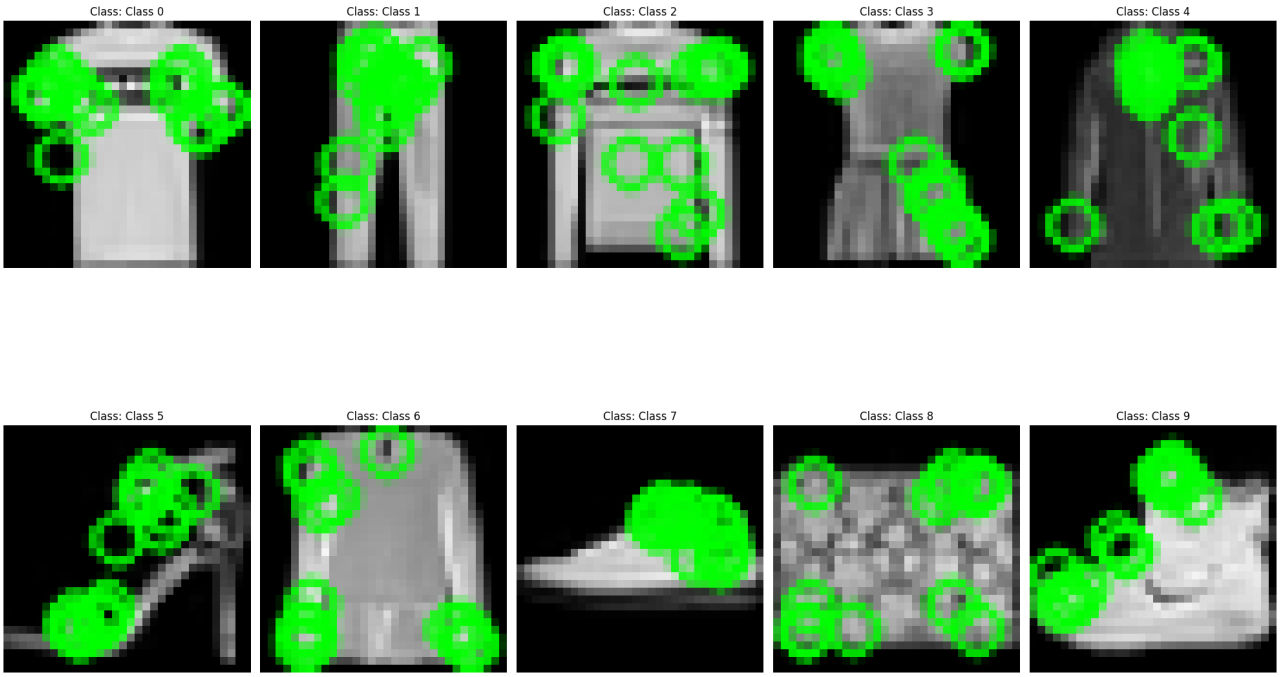
VIZUALIZARE CANTITATIVA SI CALITATIVA

FASHION

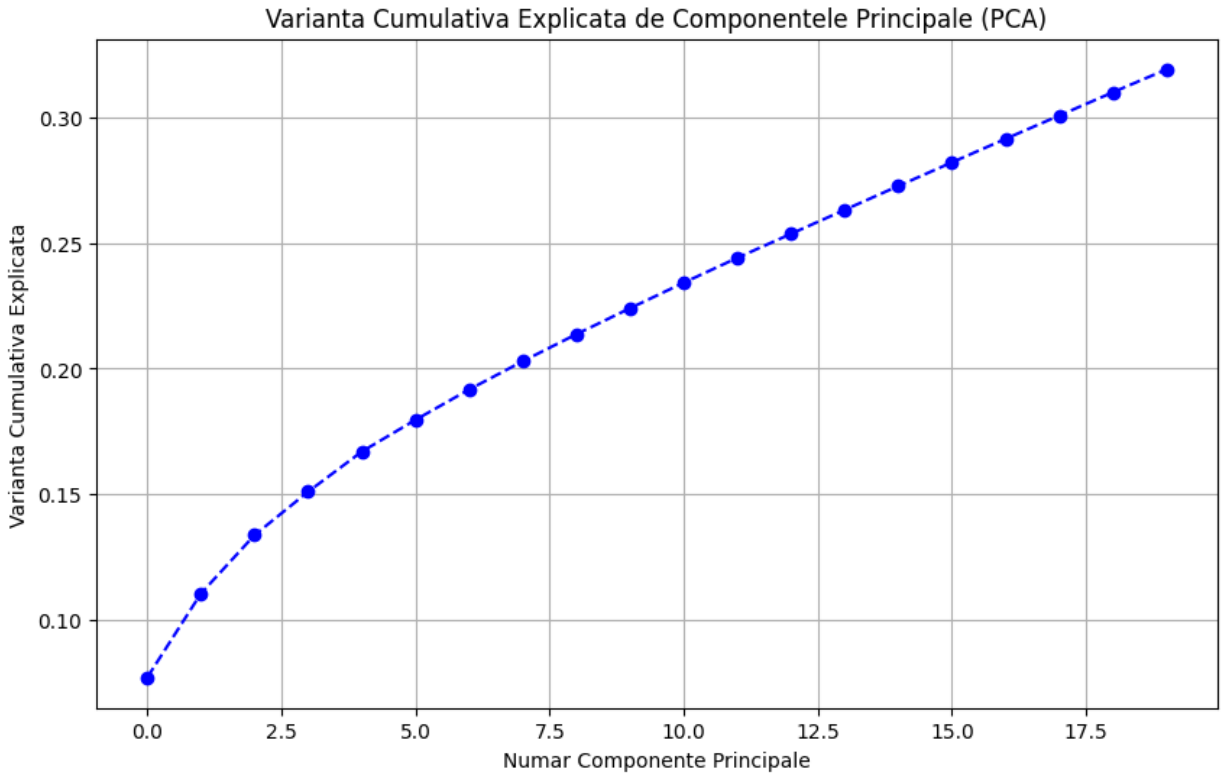


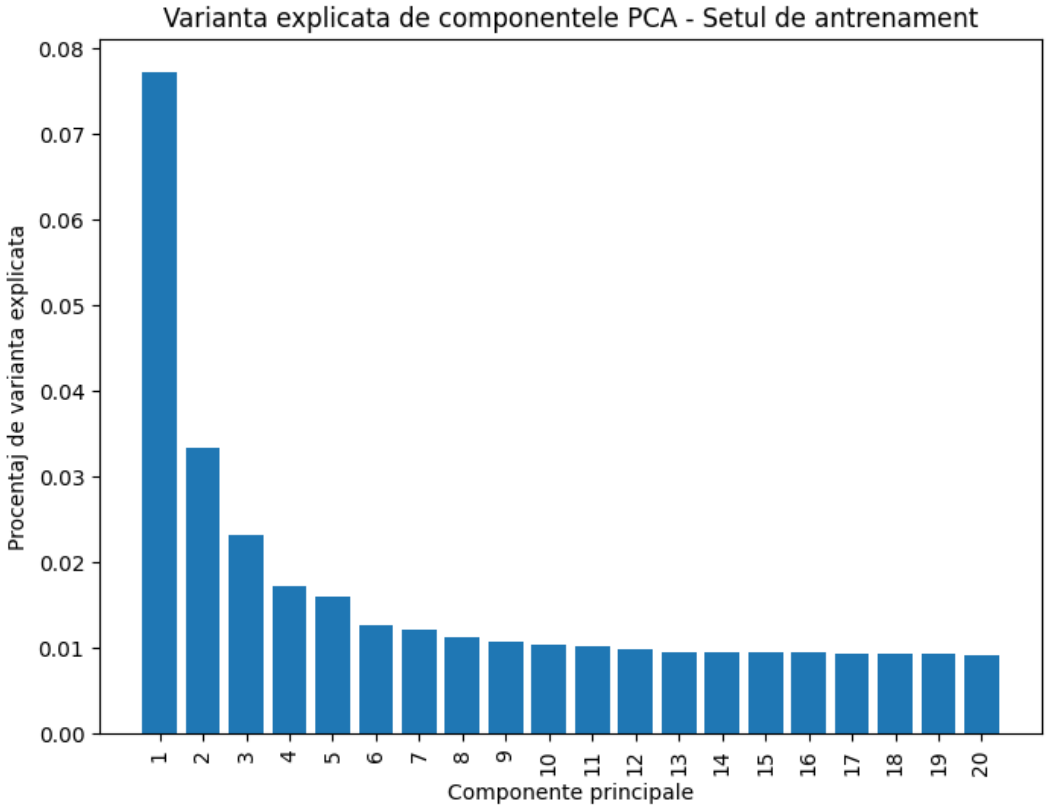
Observăm că prima componentă principală explică o mare parte din varianța datelor, urmată de componentele a doua și a treia care încă contribuie semnificativ, dar ponderea lor scade rapid. Restul componentelor explică doar o fracțiune foarte mică din varianța totală.

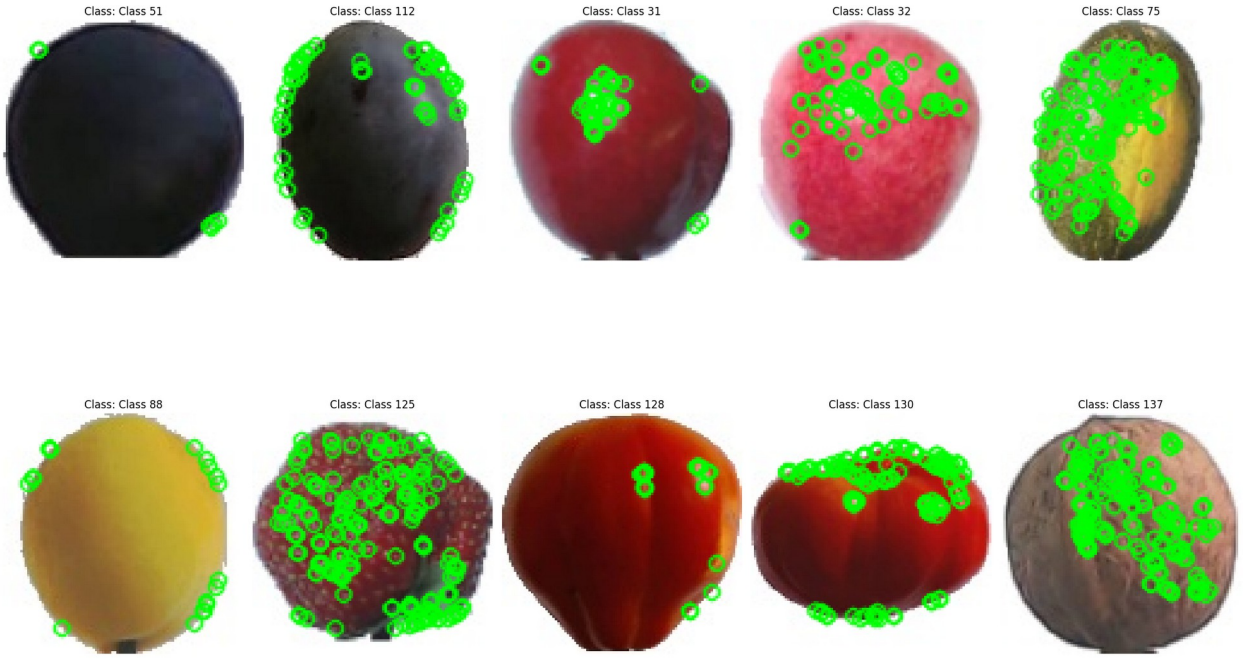



Class: Class 0
Class: Class 1
Class: Class 2
Class: Class 3
Class: Class 4
Class: Class 5
Class: Class 6
Class: Class 7
Class: Class 8
Class: Class 9

FRUITS


Varianta Cumulativa Explicata de Componentele Principale (PCA)
Varianta Cumulativa Explicata
0.10
0.15
0.20
0.25
0.30
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Numar Componente Principale


Varianta explicata de componentele PCA - Setul de antrenament
Procentaj de varianta explicata
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Componente principale


Class: Class 51
Class: Class 112
Class: Class 31
Class: Class 32
Class: Class 75
Class: Class 88
Class: Class 125
Class: Class 128
Class: Class 130
Class: Class 137

ANTRENARE

Impactul hiper-parametrilor asupra performanţei:

- **Logistic Regression**: hiperparametrul 'C' (penalizarea) a avut un impact semnificativ asupra performanței, regândirea acestuia având efect asupra stabilităţii modelului.
- **SVM**: kernel-ul ales (linear, rbf) influențează considerabil performanța, iar pentru datele non-liniare, kernel-ul rbf a oferit cele mai bune rezultate.
- **Random Forest**: numărul de arbori ('n_estimators') și adâncimea maximă ('max_depth') au fost esențiale pentru reducerea overfitting-ului și îmbunătățirea generalizării.
- **XGBoost**: parametrii de adâncime şi rata de învățare au fost foarte sensibili, iar ajustarea acestora a dus la o performanță considerabil mai bună

Atributele cele mai predictive:

- În general, **componenta principală** obținută prin PCA a fost foarte utilă în reducerea dimensiunii setului de date fără a sacrifica prea mult din informațiile esențiale. Atributele extrase prin ORB, care reprezintă texturi şi contururi ale imaginilor, au avut un impact semnificativ în predicțiile modelelor de clasificare.

- Clasele cu mai multe exemple în setul de antrenament au fost, în general, mai bine prezise. Clasele mai echilibrate au fost, de asemenea, mai uşor de prezis.

În concluzie, Logistic Regression şi XGBoost s-au dovedit a fi cele mai eficiente în cazul seturilor de date testate, cu un impact semnificativ al ajustării hiper-parametrilor asupra performanţei fiecărui model.