Glodea Eusebiu

Feature extraction, standardizare de atribute și utilizarea algoritmilor de învățare automată pe seturi de date

Proiectul are ca scop vizualizarea și "explorarea" datelor unei probleme (Exploratory Data Analysis), extragerea atributelor datelor problemei pentru a fi utilizate în obiectivul de analiză ales (e.g. clasificare, regresie, detectie de anomalii) și evaluarea mai multor modele pentru găsirea soluției celei mai bune pentru problema dată. Sunt prezentate biblioteci de vizualizare a datelor și extragerea de atribute pentru folosirea a 4 algoritmilor de clasificare.

Sunt utilizate două seturi de date cu imagini deja împărțite în date de antrenare (train) și de testare (test):

- Fashion-MNIST- un set de date cu 70000 de imagini grayscale de tip thumbnail (32 x 32) reprezentând 10 tipuri de elemente de vestimentație (e.g. bluze, cizme, pantaloni)

link: GitHub - zalandoresearch/fashion-mnist: A MNIST-like fashion product database. Benchmark

- Fruits-360- un set de date cu ~55000 de imagini RGB cu 80 de tipuri diferite de fructe singulare

link: Fruits-360 dataset

I. Extragerea de atribute

a) Setul de date Fashion

Pentru extragerea atributelor setului de date Fashion, am ales să utilizez metodele HOG și PCA.

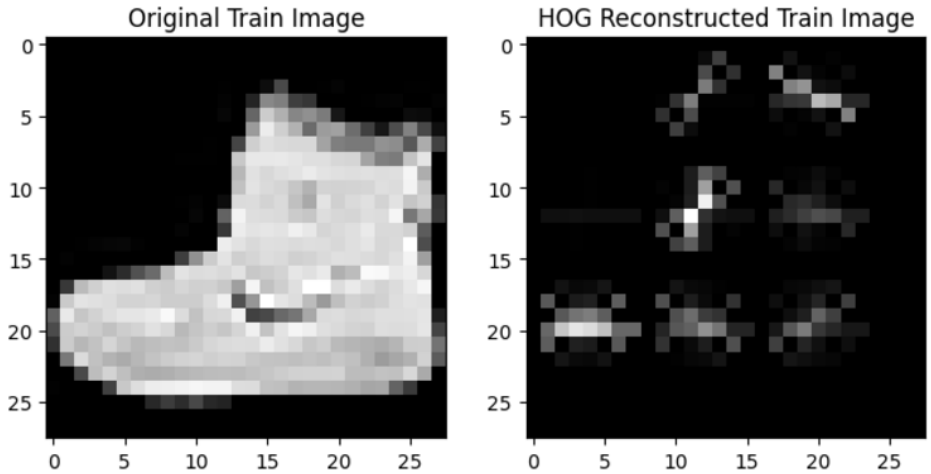
- **HOG**

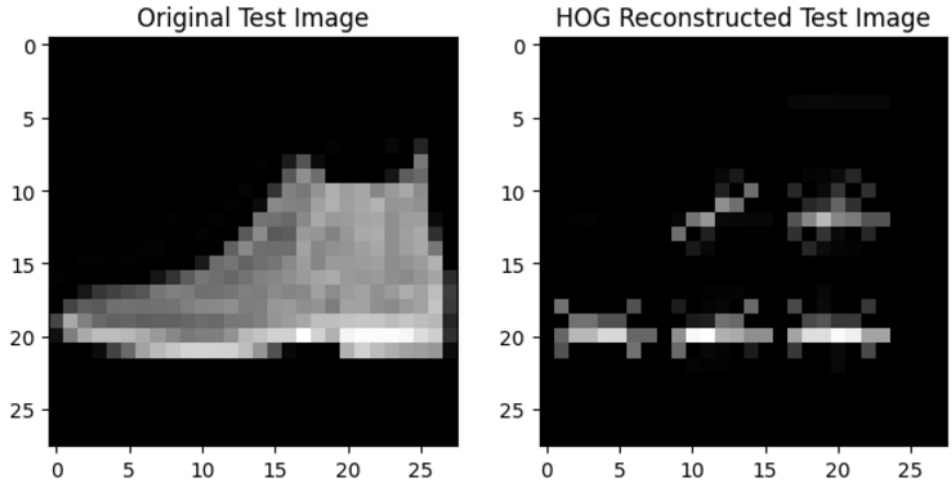
Metoda HOG a fost selectată deoarece este bună pentru captarea texturilor și a formelor din haine. Are, totuși, o sensibilitate redusă pentru analiza detaliilor mici, cum ar fi texturile fine, dar dată fiind structura imaginilor din setul de date fashion (imagini care au culori in format alb-negru și nu sunt foarte detaliate), am considerat că este o opțiune bună pentru setul de date.

Transformarea HOG se bazează pe anumiți parametrii care influențează rezultatul obținut.

Am ales orientations = 9 (9 intervale egale de grupare a unghiurilor gradientului histogramei), pixels_per_cell = (8, 8) (dimensiunea celulelor pe care se calculează histograma gradienților), cells_per_block = (2, 2) (numărul de celule dintr-un bloc utilizat pentru normalizarea pe blocuri de celule) și block_norm = L2-Hys (tipul de normalizare aplicat vectorilor de caracteristici pentru fiecare bloc de celule).

Mai jos se pot observa exemple de rezultate ale aplicării metodei HOG cu parametrii specificați mai sus pe seturile de testare si antrenare.

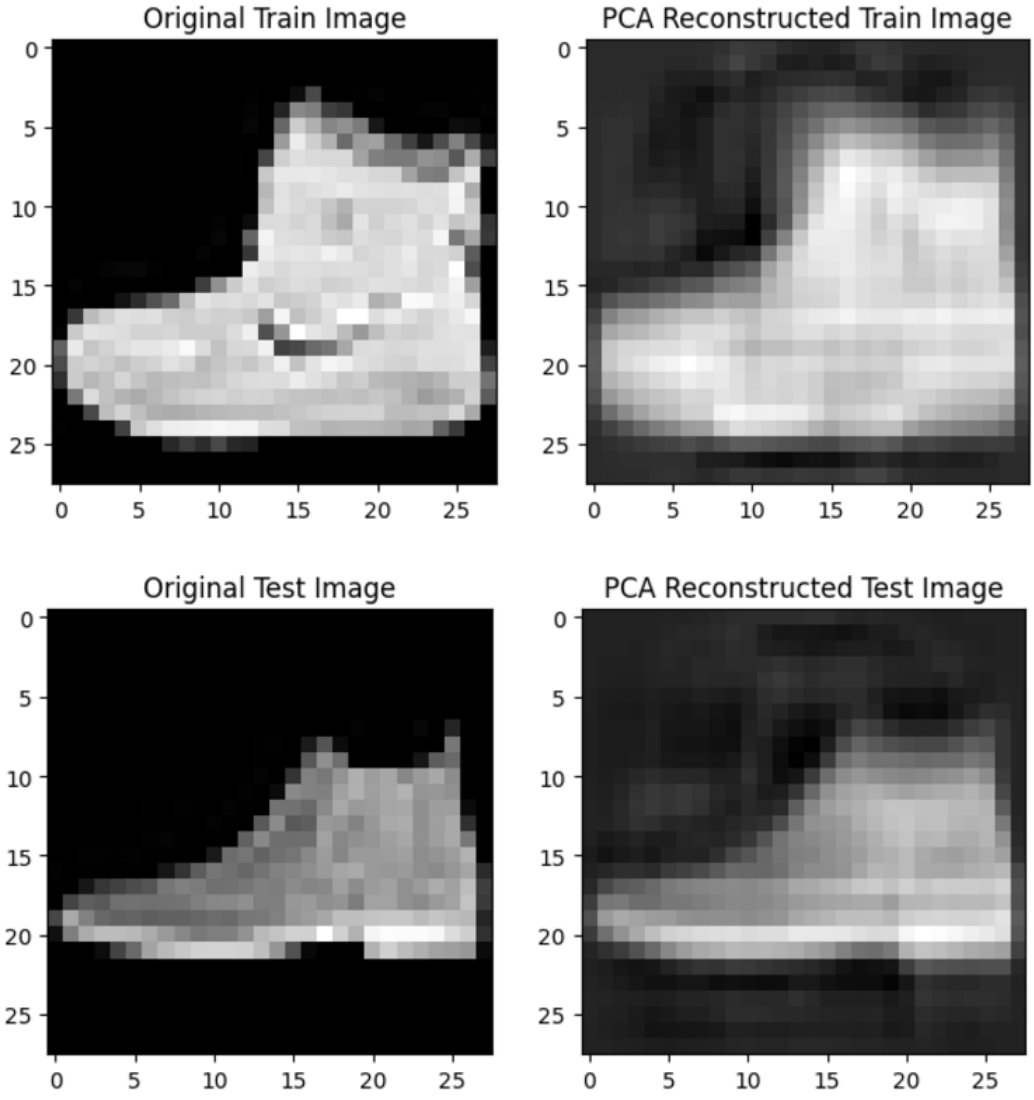




- **PCA**

Metoda PCA a fost aleasă întrucât ajută la reducerea dimensionalității unui set mare de imagini și păstrează variația majoră, fiind ideal pentru forme generale. Există posibilitatea pierderii unor informații, mai ales când se selectează doar câteva componente principale. În acest sens, am ales ca numărul de componente principale să fie de 50, un număr suficient pentru a păstra structura semnificativă a datelor, păstrându-se un procent de 90-95% din variația totală. Astfel, datele devin mai ușor de gestionat, iar algoritmii de învățare automată rulează mai rapid.

Mai jos se pot observa exemple de rezultate ale aplicării metodei PCA pe seturile de testare si antrenare.



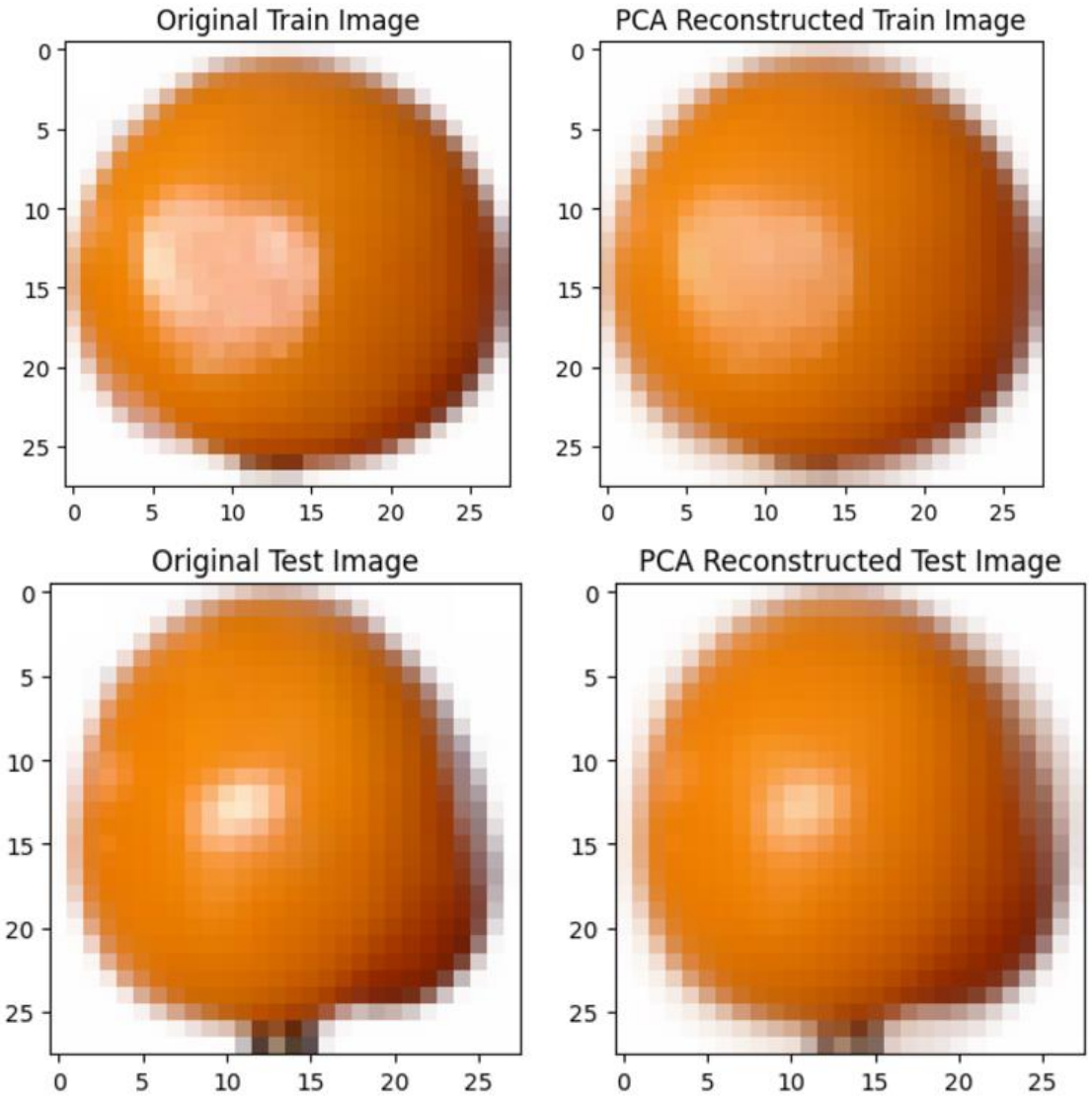
b) Setul de date Fruits

Pentru extragerea atributelor setului de date Fashion, am ales să utilizez metodele PCA și ORB.

- **PCA**

Ca și în cazul setului de date Fashion, metoda PCA este utilizată pentru a se reduce dimensionalitatea setului de date, și a se păstra variația totală a informațiilor prin alegerea tot a 50 de componente. De asemenea, în plus față de cazul setului de date Fashion, necesitatea utilizării unor metode de reducere a dimensionalității datelor este și mai mare, având în vedere faptul că setul de date Fruits are o mărime mult mai mare.

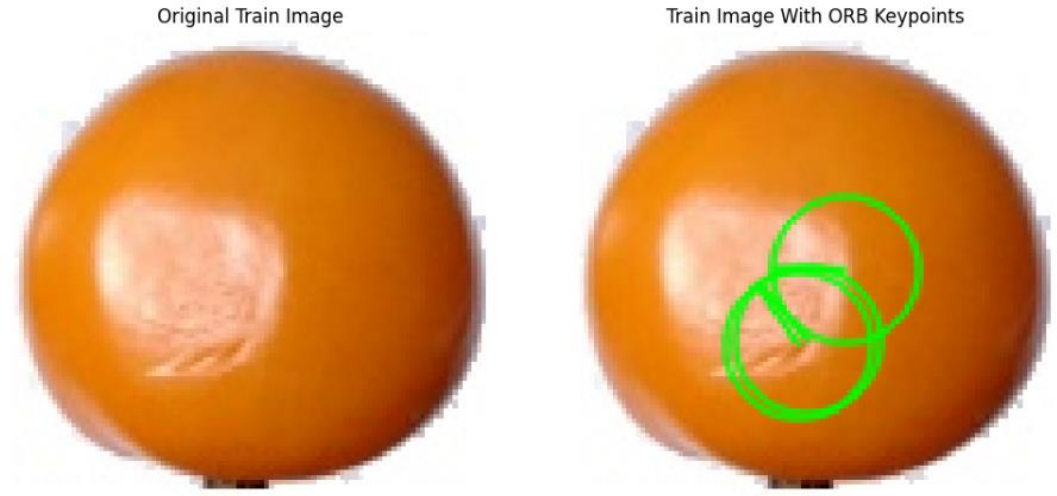
Mai jos se pot observa exemple de rezultate ale aplicării metodei PCA pe seturile de testare si antrenare. Spre deosebire de setul de date Fashion, diferențele față de imaginile originale sunt mult mai puțin observabile, lucru datorat de faptul că imaginile din Fruits sunt mult mai detaliate (și color).



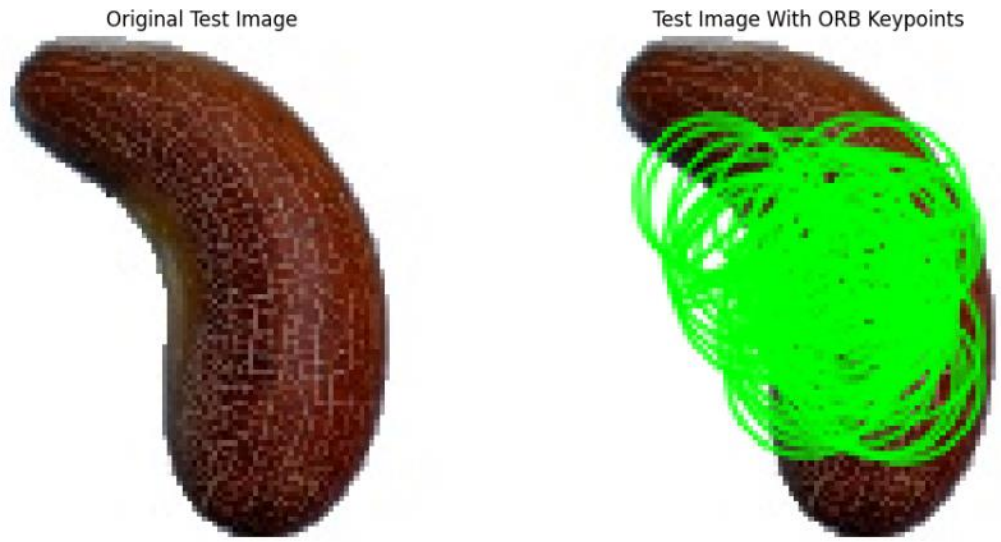
- **ORB**

Metoda ORB a fost selectată întrucât este utilă pentru detectarea detaliilor specifice ale imaginilor și, chit că nu funcționează bine cu imagini neclare sau aglomerate, în cazul setului de date Fruits nu este întampinată această problemă, fiind un set de date cu imagini foarte clare si detaliate.

Mai jos se pot observa exemple de rezultate ale aplicării metodei ORB pe seturile de testare și antrenare. Cu cât mai multe detalii semnificative există, cu atât mai multe puncte cheie vor fi identificate.




```
Number of keypoints detected: 3
```

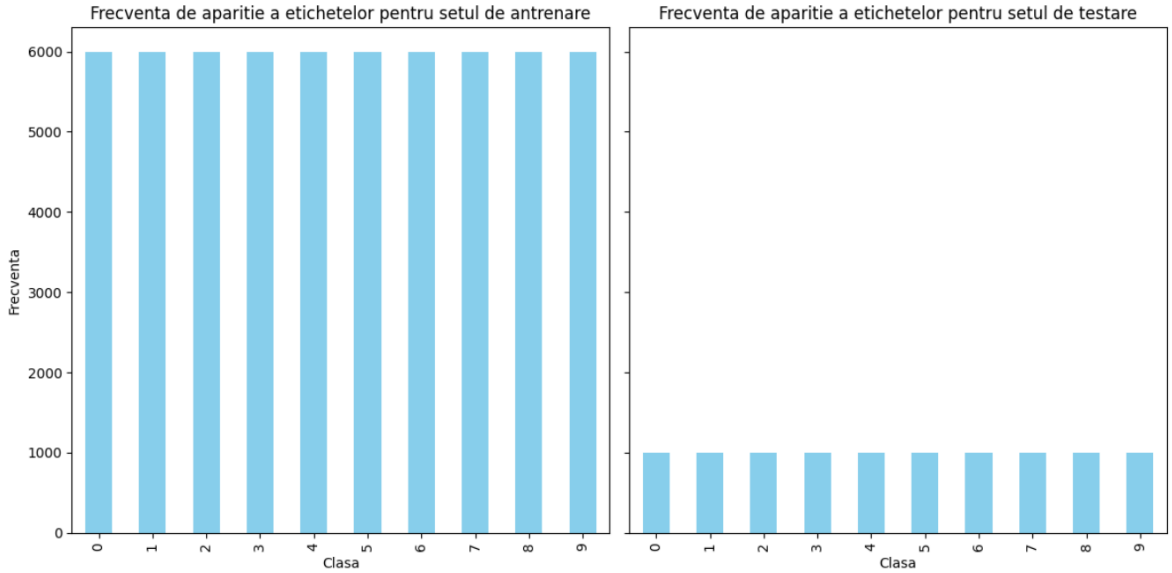



```
Number of keypoints detected: 52
```

II. Vizualizarea atributelor extrase

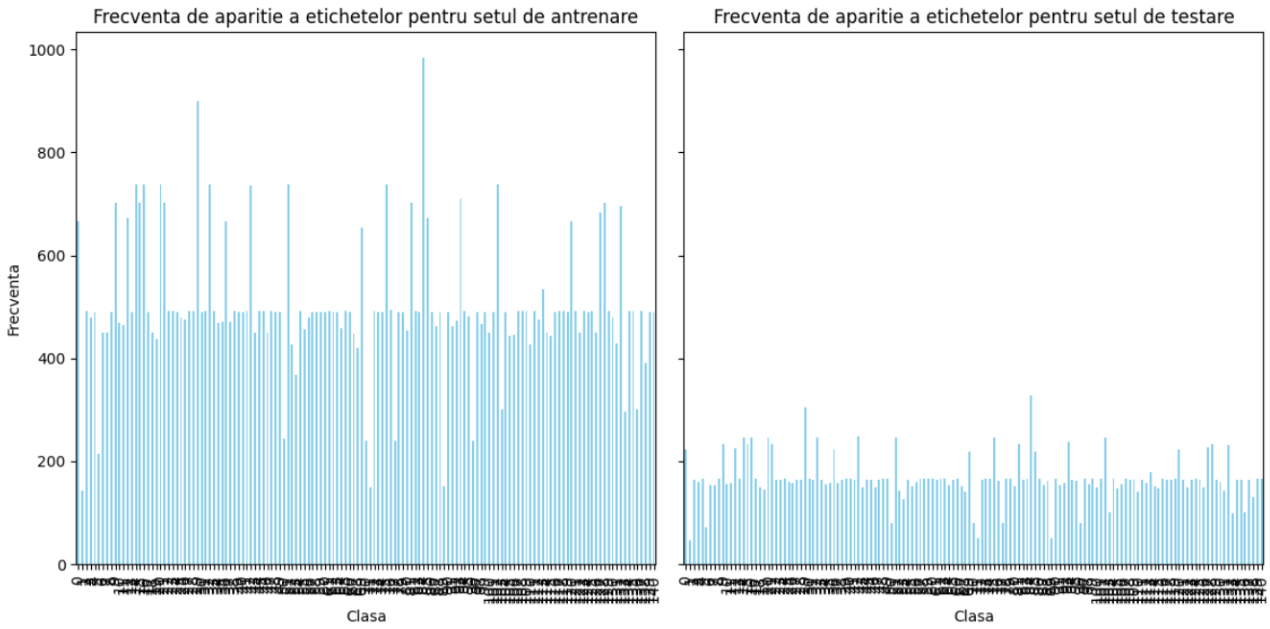
1. Analiza echilibrului de clase

a) Setul de date Fashion



Se poate observa că sunt câte 6000 de etichete (labels) pentru fiecare clasa (0-9) din setul de date de antrenare, respectiv câte 1000 de etichete pentru setul de date de testare.

b) Setul de date Fruits

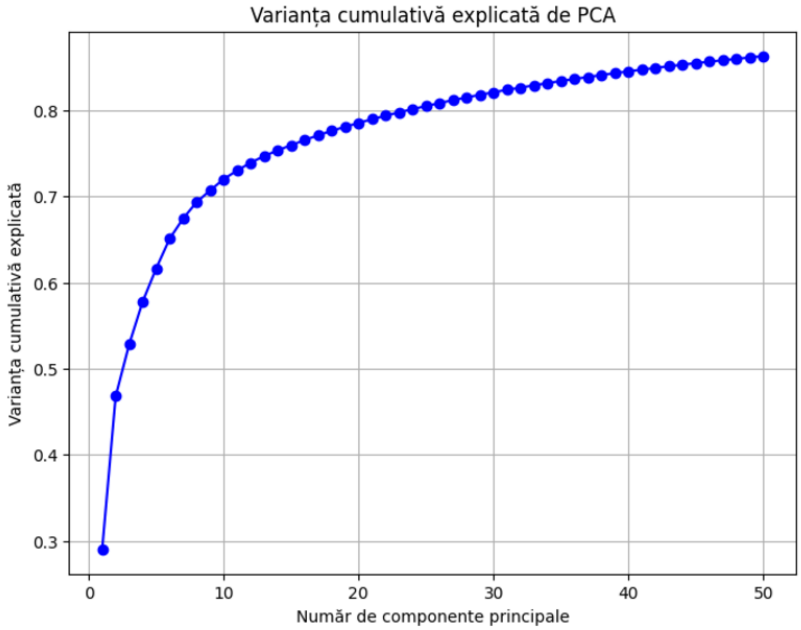


Pentru setul de antrenare, clasa cu cea mai mare frecvență este 84 (cu frecvența de 984), iar clasa cu cea mai mică frecvență este 1 (cu frecvența de 144).

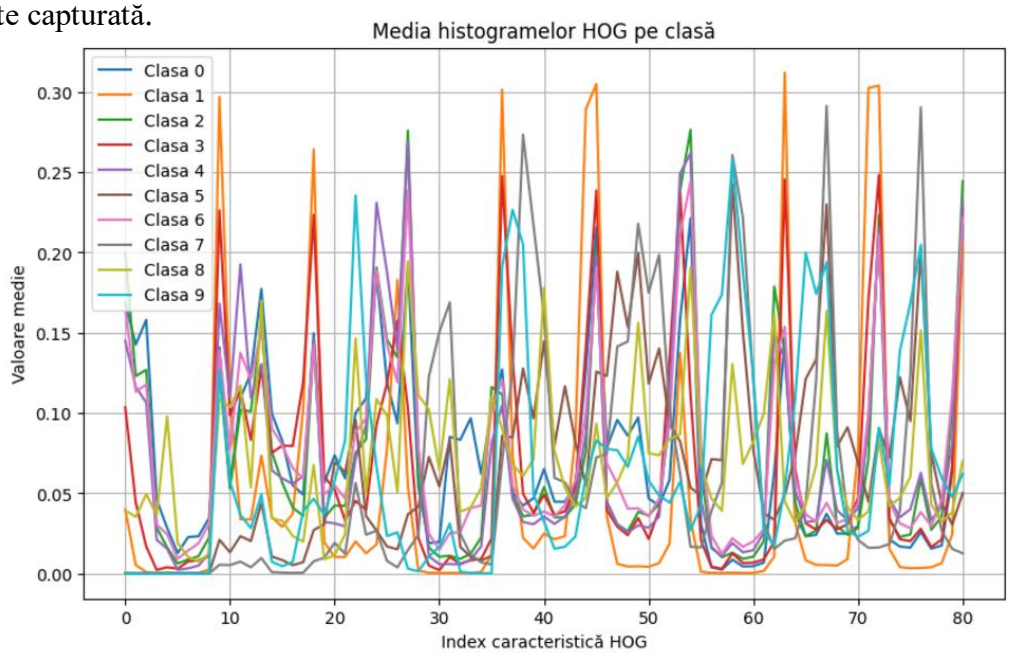
Pentru setul de testare, clasa cu cea mai mare frecvență este 84 (cu frecvența de 328), iar clasa cu cea mai mica frecvență este 1 (cu frecvența de 47).

2. Vizualizarea cantitativă a efectului de extragere a atributelor

a) Setul de date Fashion

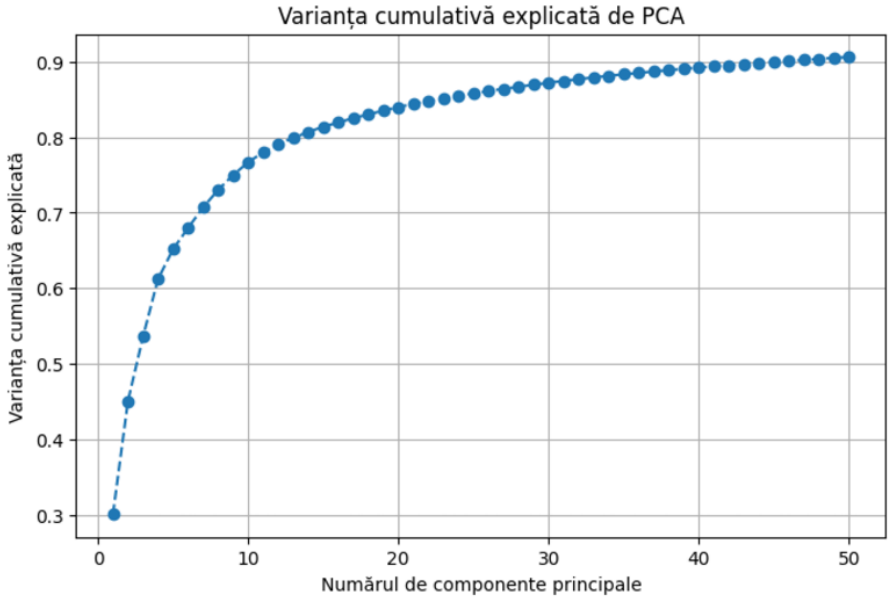


Graficul prezintă variația varianței explicate pe măsură ce se adaugă componente în cadrul metodei PCA. Se poate observa că, pe măsură ce se adaugă componente, varianța cumulativă crește. În primele 5-10 componente rata de creștere este foarte mare, dar scade pe măsură ce adăugarea de componente noi contribuie din ce în ce mai puțin la explicarea varianței. Spre 50 de componente se poate observa deja aproprierea curbei de un platou, indicând că aproape toată varianța relevantă din date este capturată.

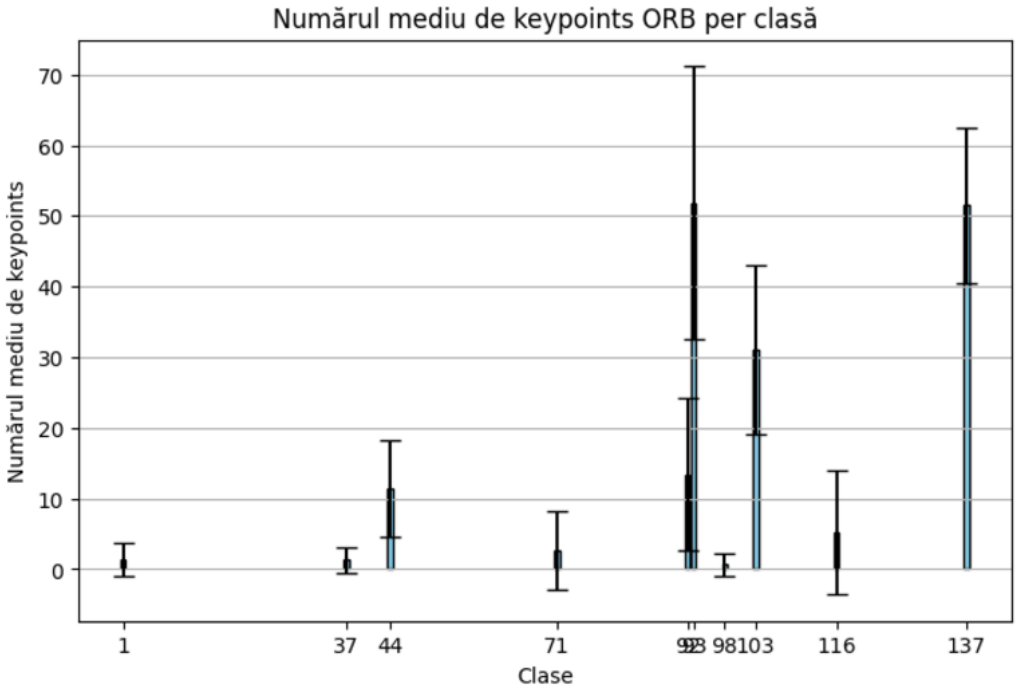


Graficul prezintă 80 de caracteristici HOG și valoarea medie a fiecărei caracteristici pe fiecare clasă de imagini. Punctele de maxim din grafic pentru anumite clase indică faptul că acele caracteristici HOG sunt foarte relevante sau frecvente în imaginile din acea clasă. În anumite intervale (cum ar fi 10-20) valorile diferitelor clase se suprapun (cum ar fi clasa 1, 3, 5 și 9) ceea ce arată faptul că acele caracteristici HOG sunt similare pentru mai multe clase, ceea ce poate face clasificarea mai dificilă.

b) Setul de date Fruits



Graficul prezintă variația varianței explicate pe măsură ce se adaugă componente în cadrul metodei PCA. Se poate observa că, pe măsură ce se adaugă componente, varianța cumulativă crește. În primele 5-10 componente rata de creștere este foarte mare, dar scade pe măsură ce adăugarea de componente noi contribuie din ce în ce mai puțin la explicarea varianței. Spre 50 de componente se poate observa deja aproprierea curbei de un platou, indicând că aproape toată varianța relevantă din date este capturată. Spre deosebire de setul de date Fashion, valoarea varianței explicată este mai mare, fiind aproximativ de 0,9, pe când la setul Fashion era de 0,8, lucru care indică faptul că metoda PCA este mai bine folosită pe setul Fruits.

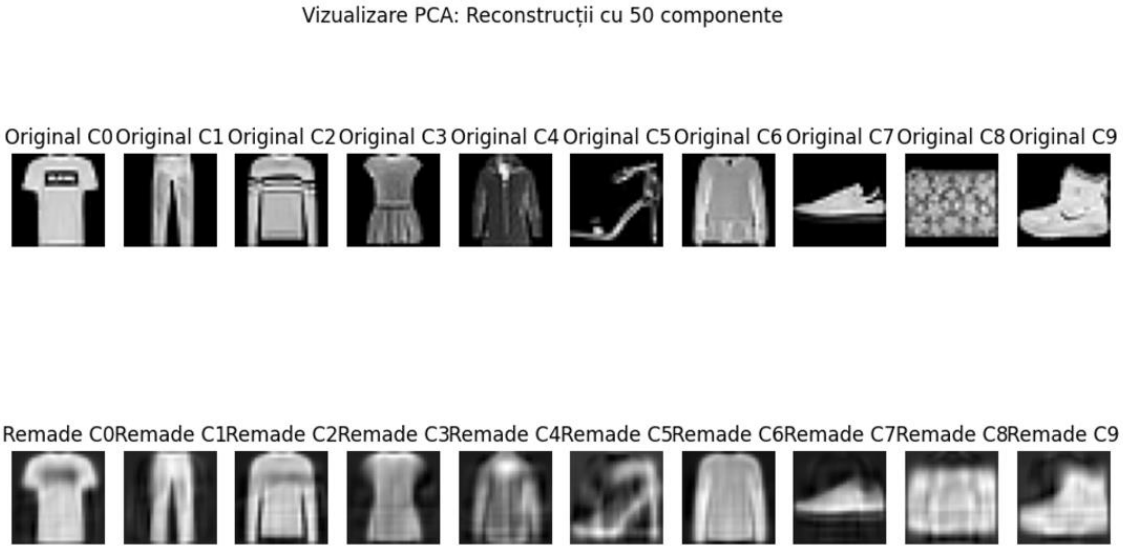


Pe grafic, barele reprezintă numărul mediu de keypoints detectate pentru imaginile din fiecare clasă, iar error bars urile indică deviația standard, adică variabilitatea numărului de keypoints între imagini din aceeași clasă. Pentru o interpretare mai bună, am extras mai jos și datele din tabel.
Clasa 1: Media keypoints = 1.37, Deviația standard = 2.39

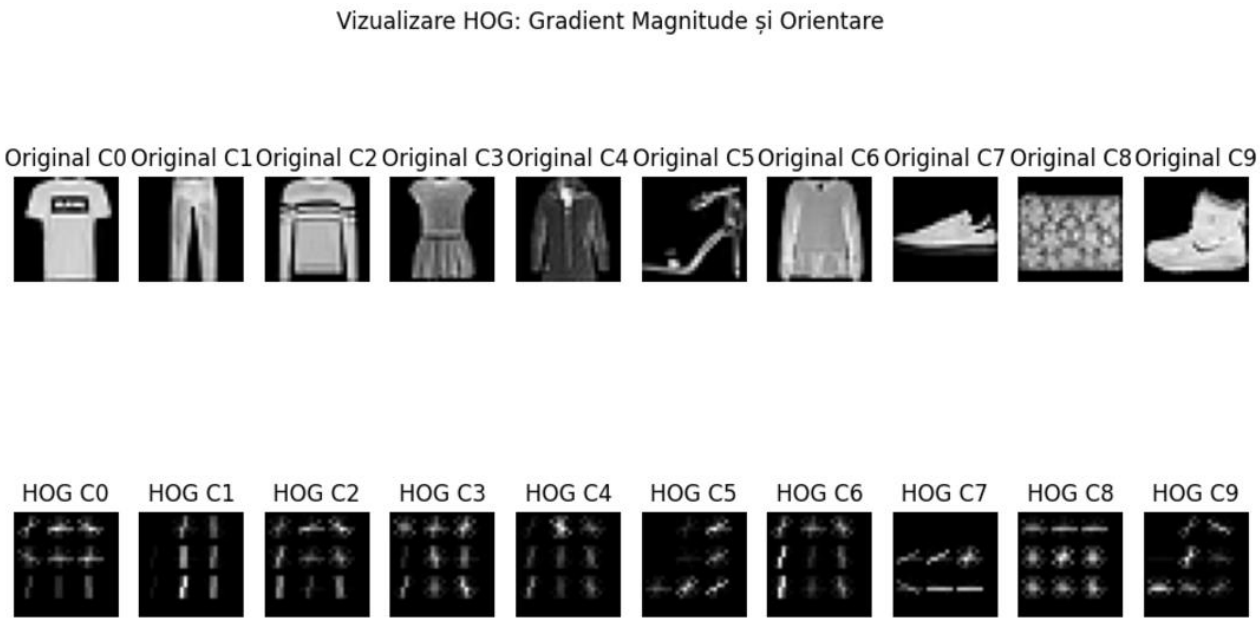
Clasa 37: Media keypoints = 1.29, Deviația standard = 1.82
Clasa 44: Media keypoints = 11.42, Deviația standard = 6.81
Clasa 71: Media keypoints = 2.70, Deviația standard = 5.61
Clasa 92: Media keypoints = 13.42, Deviația standard = 10.78
Clasa 93: Media keypoints = 51.86, Deviația standard = 19.32
Clasa 98: Media keypoints = 0.62, Deviația standard = 1.66
Clasa 103: Media keypoints = 31.14, Deviația standard = 11.96
Clasa 116: Media keypoints = 5.23, Deviația standard = 8.82
Clasa 137: Media keypoints = 51.51, Deviația standard = 11.01

3. Vizualizarea calitativă a efectului de extragere a atributelor

a) Setul de date Fashion

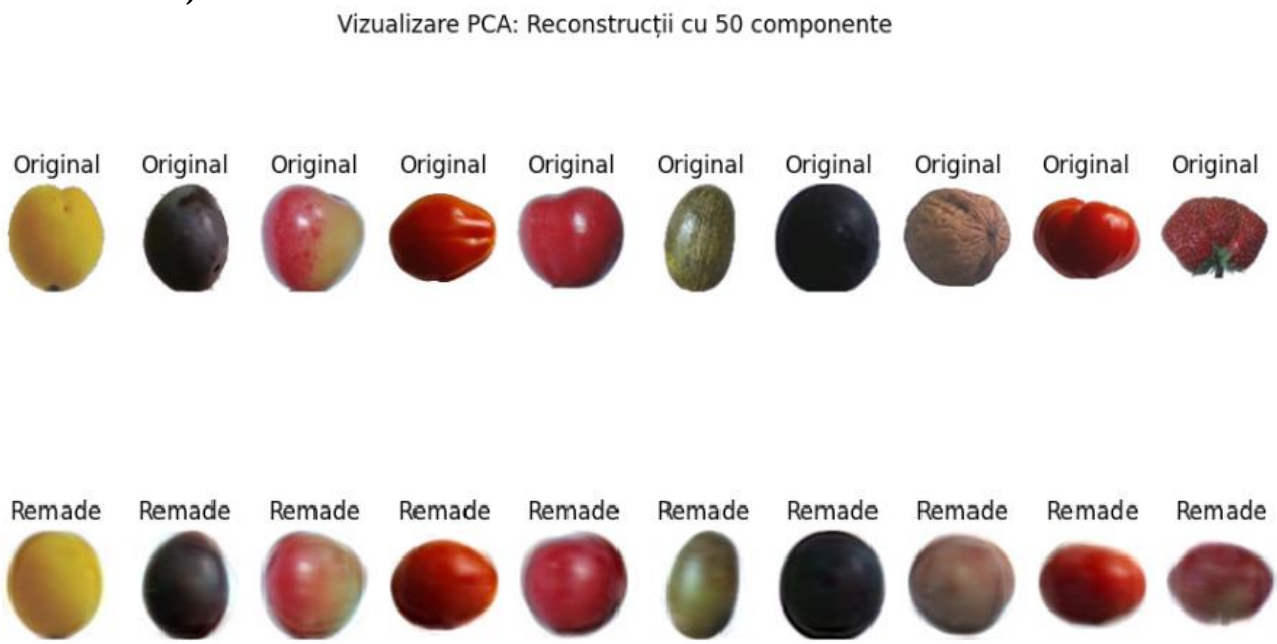


Pentru reconstruirea imaginilor PCA, se utilizează doar componentele principale pentru a se observa calitatea reconstrucției și câtă informație se pierde. Imaginile au un grad de varianță de aproximativ 90-95% față de imaginile originale, folosindu-se 50 de componente.

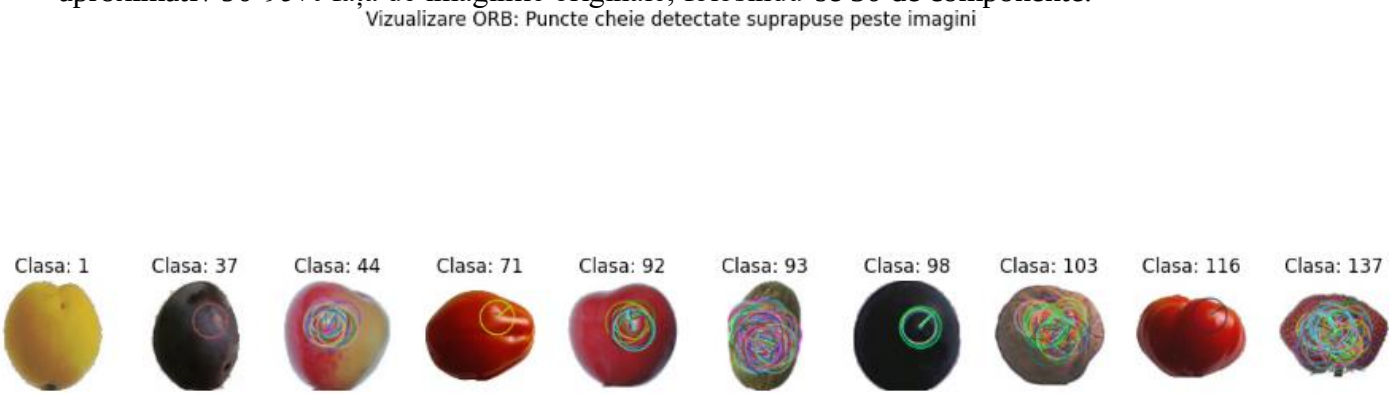


Se reconstruiesc imaginile folosindu-se metoda HOG. Imaginile inițiale, care aveau 28 x 28 pixeli = 784 componente vor avea după transformare doar 144 de componente. Fiecare bloc va genera 36 de componente (4 celule per bloc x 9 orientări), iar numărul de blocuri rezultat va fi doar 4 (2 x 2 grid de blocuri), deci 4 x 9 x 4 = 144 de componente.

b) Setul de date Fruits



Pentru reconstruirea imaginilor PCA, se utilizează doar componentele principale pentru a se observa calitatea reconstrucției și câtă informație se pierde. Imaginile au un grad de varianță de aproximativ 90-95% față de imaginile originale, folosindu-se 50 de componente.



Clasa '1' - Număr de keypoints detectate: 0, rezultat care poate apărea deoarece este fie o imagine foarte uniformă, fără detalii, fie deoarece poate avea zgomot mare, iar algoritmul nu detectează puncte de interes.
Clasa '37' - Număr de keypoints detectate: 1
Clasa '44' - Număr de keypoints detectate: 11
Clasa '71' - Număr de keypoints detectate: 1
Clasa '92' - Număr de keypoints detectate: 9
Clasa '93' - Număr de keypoints detectate: 68
Clasa '98' - Număr de keypoints detectate: 3
Clasa '103' - Număr de keypoints detectate: 28
Clasa '116' - Număr de keypoints detectate: 1
Clasa '137' - Număr de keypoints detectate: 43

III. Standardizarea și selecția atributelor

a) Setul de date Fashion

Am decis să folosesc StandardScaler() pentru a standardiza datele si SelectPercentile() pentru a selecta cele mai bune 20% caracteristici din setul de date. În mod evident, setul de date obținut în urma acestor transformări va avea mai puține atribute. Lucru care este cauzat și de aplicarea metodelor de extragere a atributelor.

b) Setul de date Fruits

De aici am întâmpinat probleme în realizarea temei. Tema am realizat-o în mediul de lucru Google Colab, iar mărimea imensă a setului de date Fruits îmi cauza rularea algoritmilor de extragere a atributelor pe întreg setul de date să cauzeze crash RAM ului. Nu a fost o soluție rularea temei pe local, tot 12GB RAM am pe local, ca și pe Google Colab.

Ce am făcut pentru a putea continua realizarea temei a fost să iau setul de date cu cele top 10 cele mai numeroase clase de la cerința cu vizualizări calitative și cantitative și să lucrez mai departe cu el. Totuși, nici procedurile de extragere de atribute nu am putut sa le extrag, ajungând să am tot runtime error, chit ca am mai puține date. Totuși, am putut să fac proceduri de standardizare pe setul de date și să lucrez mai departe cu el.

Ca și în cazul setului de date Fashion, am ales StandardScaler() pentru a standardiza datele, și SelectPercentile() pentru a selecta cele mai bune 5% caracteristici din setul de date. De asemenea, pentru a elimina din atributele constante (cu varianța 0), am aplicat VarianceTreshold() pe setul de date. Și în cazul acesta, în urma transformărilor menționate, setul de date va avea mai puține atribute și este pregătit de clasificare.

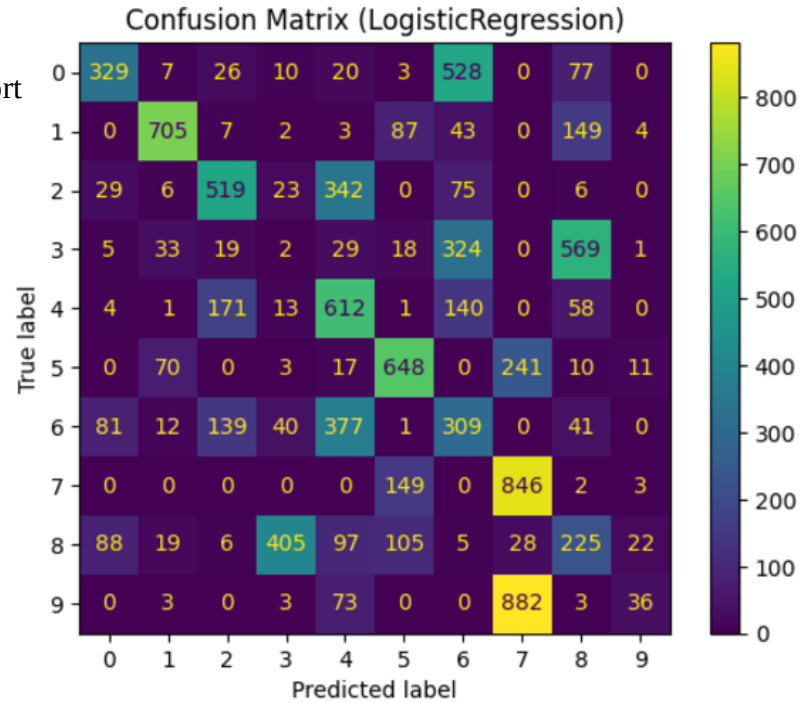
IV. Utilizarea Algoritmilor de Învățare Automată

a) Setul de date Fashion

- **Regresie Logistică**

Best parameters: {'C': 10, 'multi_class': 'multinomial'

	precision	recall	f1-score	support
0	0.61	0.33	0.43	1000
1	**0.82**	0.70	**0.76**	1000
2	0.59	0.52	0.55	1000
3	0.00	0.00	0.00	1000
4	0.39	0.61	0.48	1000
5	0.64	0.65	0.64	1000
6	0.22	0.31	0.25	1000
7	0.42	**0.85**	0.56	1000
8	0.20	0.23	0.21	1000
9	0.47	0.04	0.07	1000
accuracy			0.42	10000

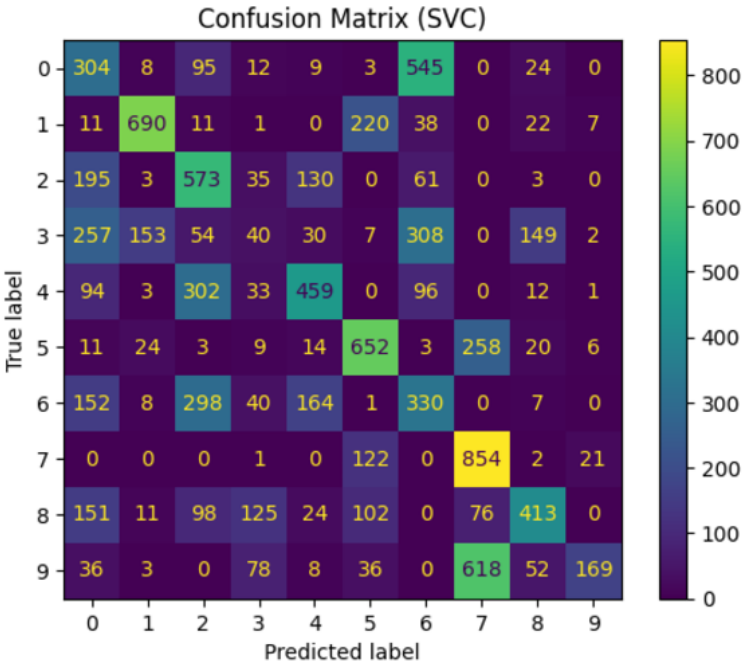


Glodea Eusebiu

- **SVM**

Best parameters: {'C': 10, 'kernel': 'rbf'}

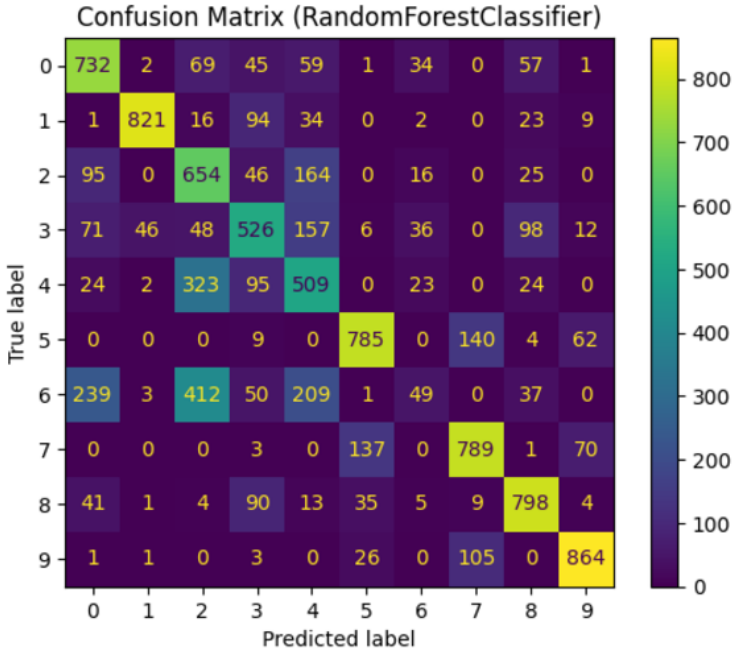
	precision	recall	f1-score	support
0	0.25	0.30	0.27	1000
1	0.76	0.69	**0.73**	1000
2	0.40	0.57	0.47	1000
3	0.11	0.04	0.06	1000
4	0.55	0.46	0.50	1000
5	0.57	0.65	0.61	1000
6	0.24	0.33	0.28	1000
7	0.47	**0.85**	0.61	1000
8	0.59	0.41	0.48	1000
9	**0.82**	0.17	0.28	1000
accuracy			0.45	10000



- **Random Forest**

Best parameters: {'max_depth': 4, 'max_features': 'sqrt', 'n_estimators': 100}

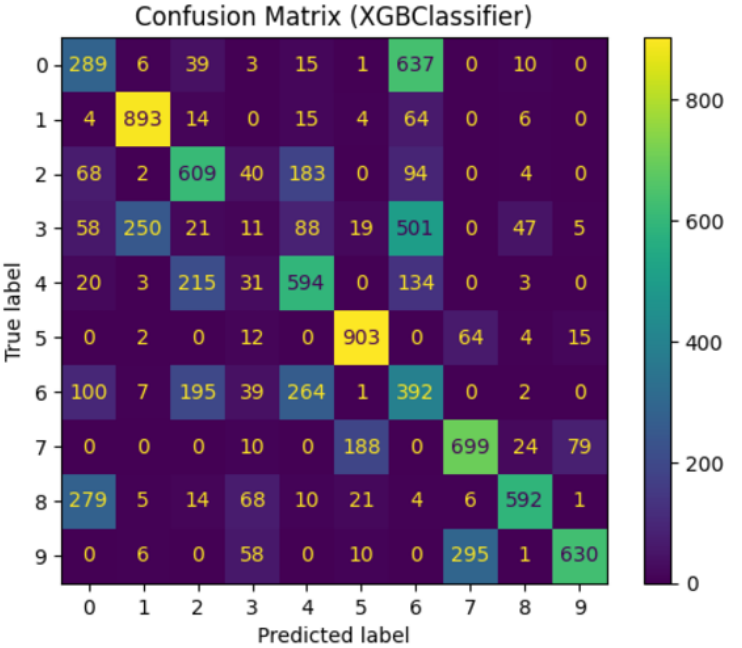
	precision	recall	f1-score	support
0	0.61	0.73	0.66	1000
1	**0.94**	0.82	**0.88**	1000
2	0.43	0.65	0.52	1000
3	0.55	0.53	0.54	1000
4	0.44	0.51	0.47	1000
5	0.79	0.79	0.79	1000
6	0.30	0.05	0.08	1000
7	0.76	0.79	0.77	1000
8	0.75	0.80	0.77	1000
9	0.85	**0.86**	0.85	1000
accuracy			0.65	10000



- **Gradient Boosted Trees**

Best parameters: {'learning_rate': 0.3, 'max_depth': 4, 'n_estimators': 100}

	precision	recall	f1-score	support
0	0.35	0.29	0.32	1000
1	0.76	0.89	0.82	1000
2	0.55	0.61	0.58	1000
3	0.04	0.01	0.02	1000
4	0.51	0.59	0.55	1000
5	0.79	**0.90**	**0.84**	1000
6	0.21	0.39	0.28	1000
7	0.66	0.70	0.68	1000
8	0.85	0.59	0.70	1000
9	**0.86**	0.63	0.73	1000
accuracy			0.56	10000

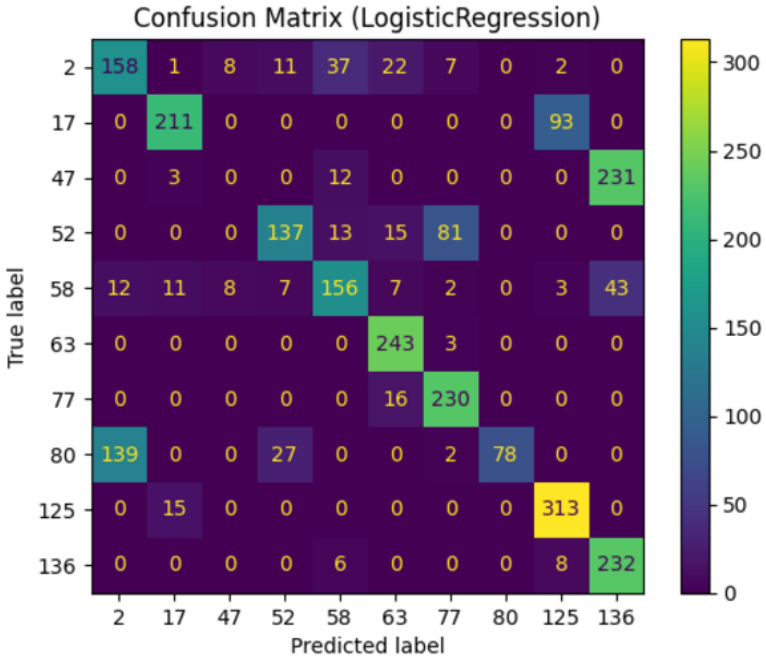


b) Setul de date Fruits

- **Regresie Logistică**

Best parameters: {'C': 10, 'multi_class': 'multinomial'}

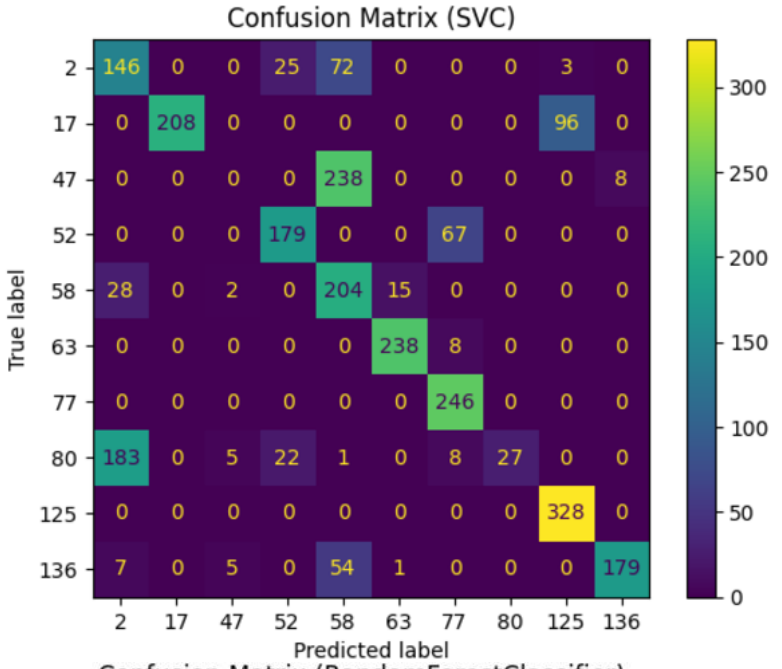
	precision	recall	f1-score	support
2	0.51	0.64	0.57	246
17	0.88	0.69	0.77	304
47	0.00	0.00	0.00	246
52	0.75	0.56	0.64	246
58	0.70	0.63	0.66	249
63	0.80	**0.99**	**0.89**	246
77	0.71	0.93	0.81	246
80	**1.00**	0.32	0.48	246
125	0.75	0.95	0.84	328
136	0.46	0.94	0.62	246
accuracy			0.68	2603



- **SVM**

Best parameters: {'C': 1, 'kernel': 'linear}

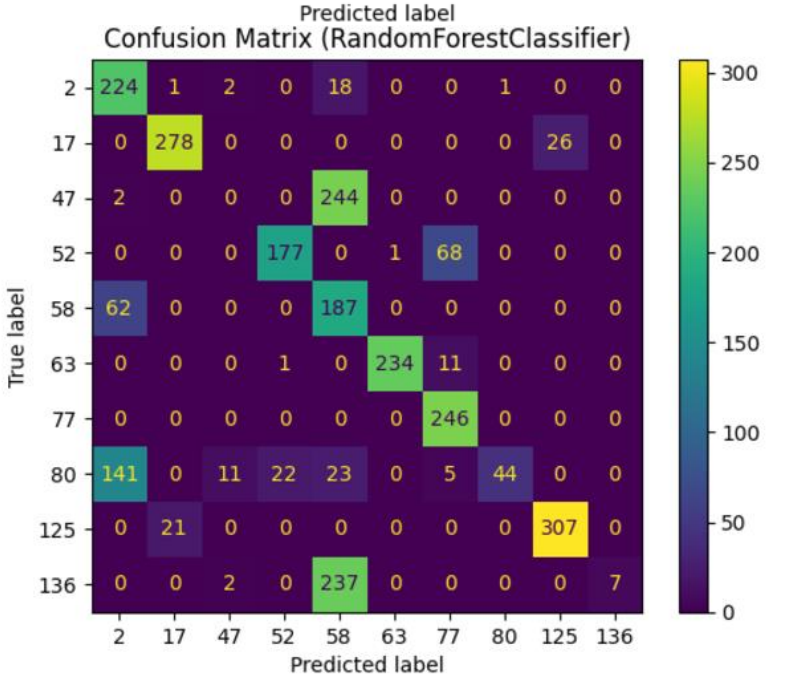
	precision	recall	f1-score	support
2	0.40	0.59	0.48	246
17	**1.00**	0.68	0.81	304
47	0.00	0.00	0.00	246
52	0.79	0.73	0.76	246
58	0.36	0.82	0.50	249
63	0.94	0.97	**0.95**	246
77	0.75	**1.00**	0.86	246
80	**1.00**	0.11	0.20	246
125	0.77	**1.00**	0.87	328
136	0.96	0.73	0.83	246
accuracy			0.67	2603



- **Random Forest**

Best parameters: {'max_depth': 15, 'max_features': 'sqrt', 'n_estimators': 100}

	precision	recall	f1-score	support
2	0.52	0.91	0.66	246
17	0.93	0.91	0.92	304
47	0.00	0.00	0.00	246
52	0.89	0.72	0.79	246
58	0.26	0.75	0.39	249
63	**1.00**	0.95	**0.97**	246
77	0.75	**1.00**	0.85	246
80	0.98	0.18	0.30	246
125	0.92	0.94	0.93	328
136	**1.00**	0.03	0.06	246
accuracy			0.65	2603



- **Gradient Boosted Trees**

Best parameters: {'learning_rate': 0.1, 'max_depth': 15, 'n_estimators': 150}

Implementarea algoritmului dn biblioteca xgboost permite utilizarea lui y_train si y_test doar cu clase ordonate în ordine numerica, (adica 0, 1, 2, ..., nu 2, 17, 47, ...), și am fost nevoit să redenumesc numele claselor din label, lucru care se poate observa pe matricea de confuzie. Pentru claritate, mai jos se află reprezentarea fiecarei clase:

Clasa 0 = 2
Clasa 1 = 17
Clasa 2 = 47
Clasa 3 = 52
Clasa 4 = 58
Clasa 5 = 63
Clasa 6 = 77
Clasa 7 = 80
Clasa 8 = 125
Clasa 9 = 136

	precision	recall	f1-score	support
2	0.39	0.78	0.52	246
17	**0.97**	0.91	0.94	304
47	0.44	0.09	0.15	246
52	0.85	0.73	0.78	246
58	0.32	0.73	0.44	249
63	0.40	0.11	0.18	246
77	0.59	**1.00**	0.74	246
80	**0.97**	0.12	0.22	246
125	0.92	0.98	**0.95**	328
136	0.41	0.20	0.26	246
accuracy			0.59	2603