

# Tema 2 ML

De Florea Radu 341C4

Aceasta parte a temei continua prima parte, unde am aplicat extragerea de atribute folosind HOG si PCA, iar clasificarea s-a facut cu metode clasice precum regresie liniara, SVM, Random Forest si Gradient Boosted Trees.

A2a parte are ca scop explorarea mai multor abordari pentru clasificarea imaginilor folosind rețele neurale. Am lucrat cu doua seturi de date, Fashion-MNIST si Fruits-360, si am testat diferite tipuri de arhitecturi, cum ar fi MLP (Multi-Layer Perceptron) si rețele convolutionale. In plus, am aplicat si o procedura de finetuning folosind o retea pre-antrenata (ResNet-18).

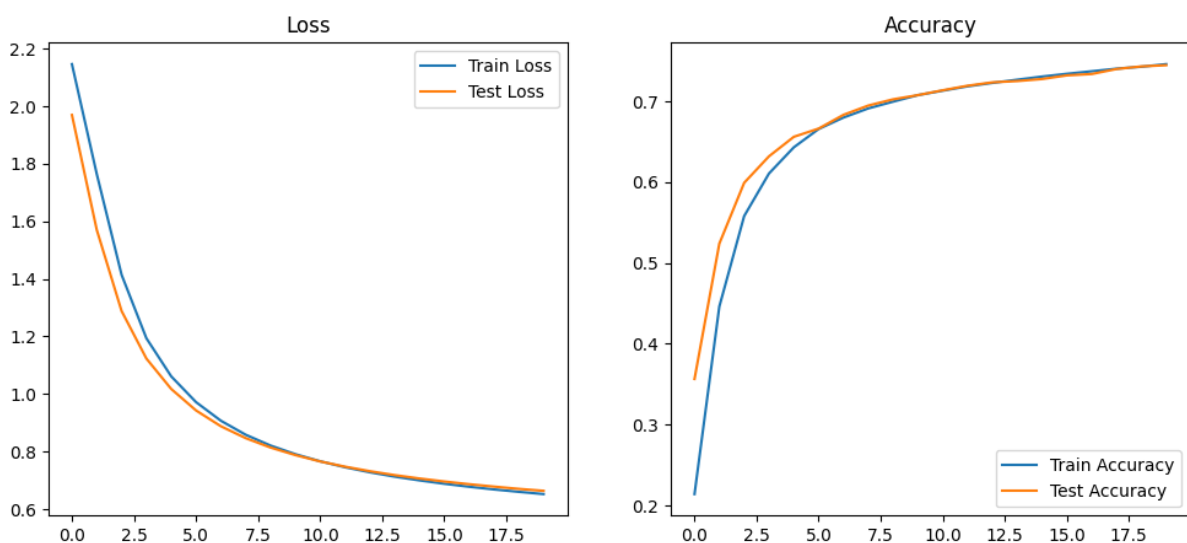
## 3.1. MLP pe atributele extrase în etapa 1

Pentru partea asta, am făcut o rețea neuronală simplă de tip MLP (Multi-Layered Perceptron) și am antrenat-o pe datele obținute din etapa 1

### Fashion

Am început prin a lua din prima parte funcțiile pentru atribute, eu am folosit pca si hog, am utilizat SelectPercentile pentru a ramane cu 58 de atribute, apoi am antrenat rețeaua neurala cu un model pe atributele rezultate.

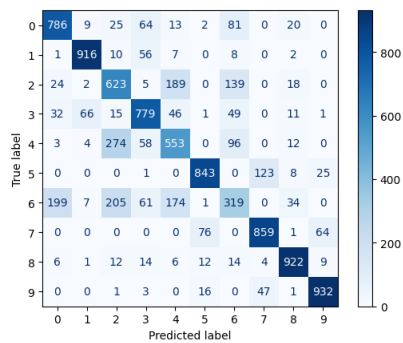
**Am antrenat pe 20 de epoci si un learning rate de 0.00001, MLP-ul are un strat de intrare (100 de neuroni), unul de iesire(10 neuroni) si unul ascuns (100 de neuroni), cu functia de activare relu intre straturi, ca functie de eroare am folosit CrossEntropyLoss, iar pentru Optimizator Adam si am rulat pe cuda (GPU)**



Loss scade constant pentru train si test, convergand spre valori apropiate . Acest lucru arata ca modelul invata eficient si generalizeaza bine fara overfitting.

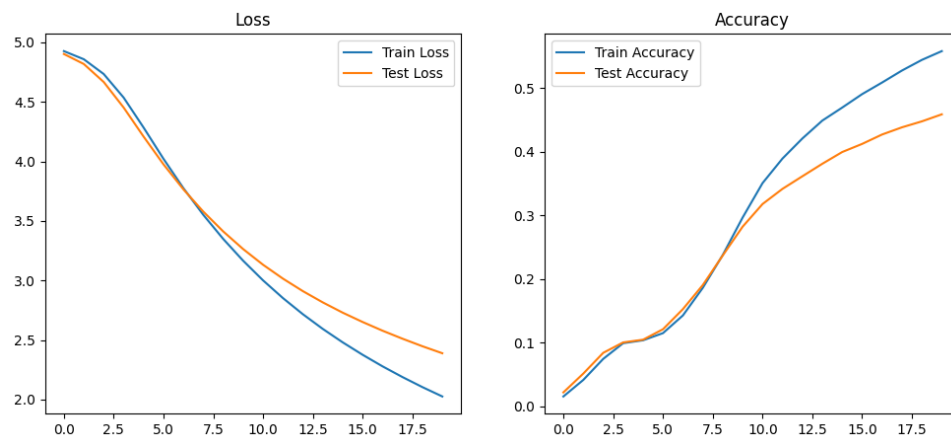
Acuratetea creste rapid la inceput, stabilizandu-se peste 70%. Apropierea valorilor pe train si test sugereaza un model echilibrat, fara overfitting.

Class	Precision	Recall	F1-Score	Support
0	0.75	0.79	0.77	1000
1	0.91	0.92	0.91	1000
2	0.53	0.62	0.58	1000
3	0.75	0.78	0.76	1000
4	0.56	0.55	0.56	1000
5	0.89	0.84	0.86	1000
6	0.45	0.32	0.37	1000
7	0.83	0.86	0.85	1000
8	0.90	0.92	0.91	1000
9	0.90	0.93	0.92	1000
accuracy		0.75	0.75	10000
macro avg	0.75	0.75	0.75	10000
weighted avg	0.75	0.75	0.75	10000



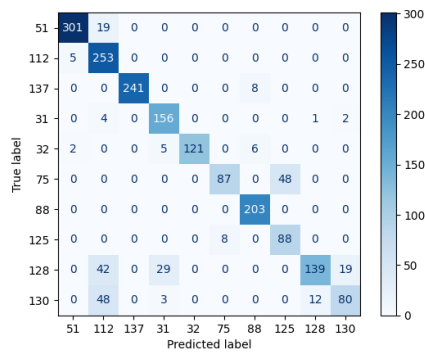
## Fruits360

Pentru Fruits360 am folosit aceeasi functie ca la fashion, dar pentru reducerea atributelor la sub 128 , am folosit VarianceThreshold



Graficul din stanga arata ca eroarea scade pentru ambele seturi, dar diferenta dintre train si test creste, semnaland un posibil overfitting. In dreapta, acuratetea creste constant, dar modelul performeaza mai bine pe train decat pe test.

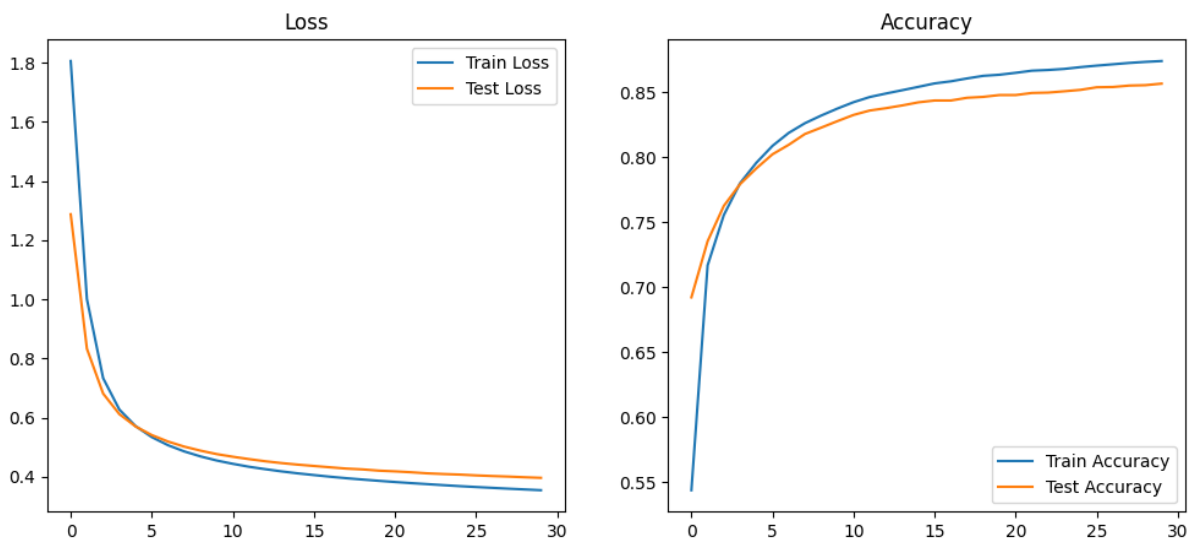
	precision	recall	f1-score	support
<b>accuracy</b>			0.46	23619
<b>macro avg</b>	0.47	0.45	0.43	23619
<b>weighted avg</b>	0.46	0.46	0.44	23619



## 3.2 Arhitectura de tip MLP direct peste imagini

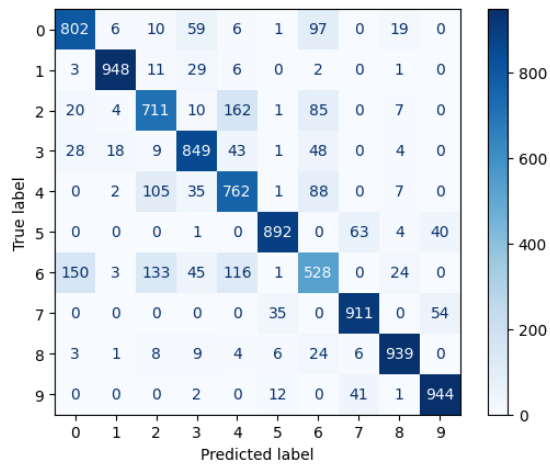
Pentru aceasta parte, am folosit exact aceeasi retea MLP ca in sectiunea 3.1. Diferenta este ca, de data aceasta, nu am mai folosit vectorii de attribute extrasi, ci am liniarizat direct imaginile din setul de date si le-am folosit ca intrare in retea. Am antrenat rețeaua timp de 30 de epoci, cu o rată de învățare mică de 0.00001. Optimizatorul folosit a fost Adam, iar funcția de eroare a fost CrossEntropyLoss

### Fashion



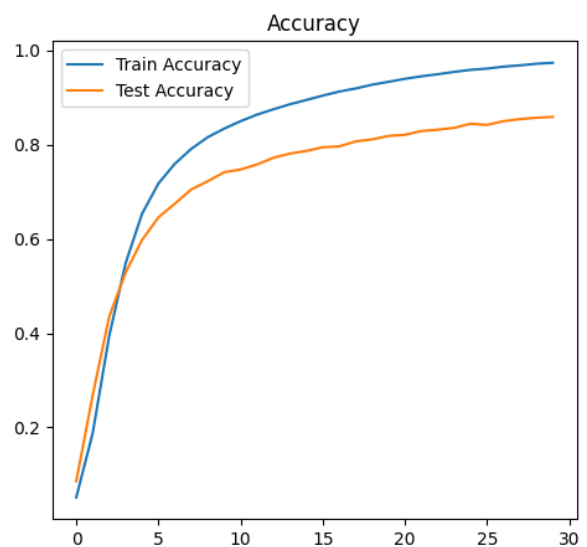
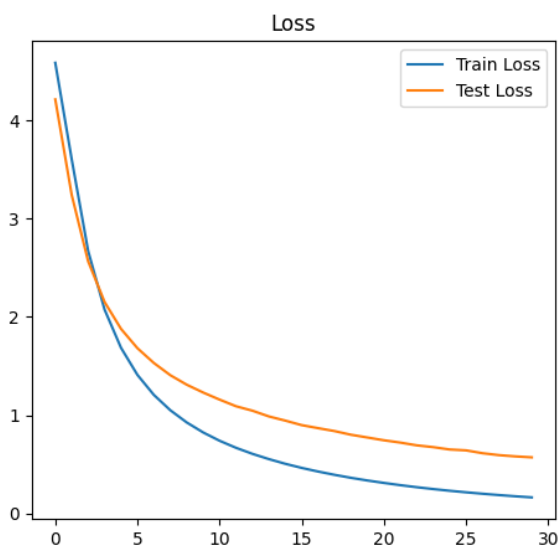
Eroarea scade constant pentru ambele seturi si se stabilizeaza aproape de 0.4, ceea ce indica o performanta buna a modelului fara semne de overfitting. Acuratetea ajunge la peste 85% pentru setul de test, cu o mica diferenta fata de train, ceea ce arata ca modelul e bun.

Class	Precision	Recall	F1-Score	Support
0	0.81	0.82	0.82	1000
1	0.98	0.95	0.97	1000
2	0.76	0.77	0.77	1000
3	0.84	0.89	0.86	1000
4	0.75	0.78	0.77	1000
5	0.94	0.91	0.92	1000
6	0.69	0.61	0.65	1000
7	0.90	0.93	0.91	1000
8	0.95	0.96	0.96	1000
9	0.93	0.94	0.93	1000
<b>accuracy</b>		0.86	0.86	10000
<b>macro avg</b>	0.86	0.86	0.86	10000
<b>weighted avg</b>	0.86	0.86	0.86	10000



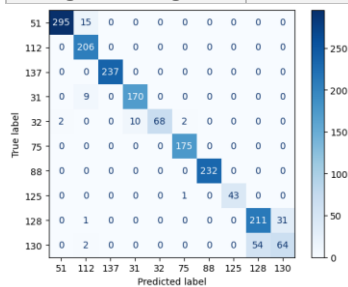
## Fruits360

Diferentele la fruits este ca pozele sunt mai mari si RGB



Eroarea scade semnificativ pentru ambele seturi, dar diferenta dintre train si test devine vizibila, sugerand un posibil overfitting. Acuratetea pe setul de test se stabilizeaza aproape de 80%, iar cea de train se apropie de 100%, ceea ce indica o generalizare mai slaba. Ar fi utila o ajustare a regularizarii.

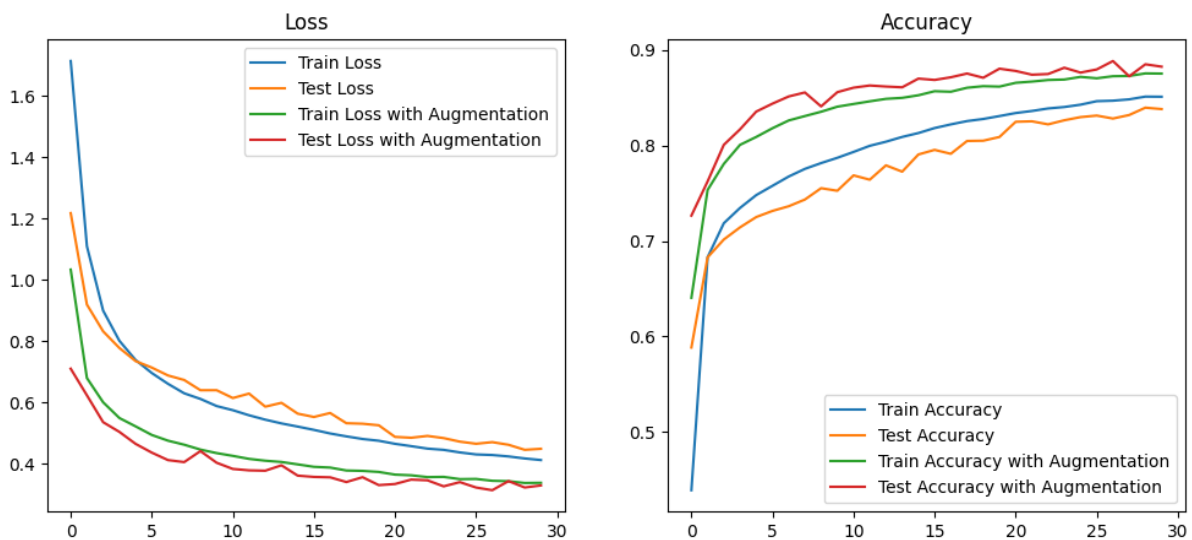
Class	Precision	Recall	F1-Score	Support
accuracy			0.87	23619
macro avg	0.88	0.87	0.87	23619
weighted avg	0.88	0.87	0.87	



### 3.3 Arhitectura de tip convoluțional

Pentru acest task, am luat clasele ConvBlock din laborator , folosind layerul BatchNorm2d si clasa DeepConvNet, unde am adaugat Dropout. Am folosit 30 de epoci. Am rulat setul de imagini cu augmentari si fara. Am folosit augmentarile din labroator: RandomHorizontalFlip, RandomCrop, si Normalizarea.

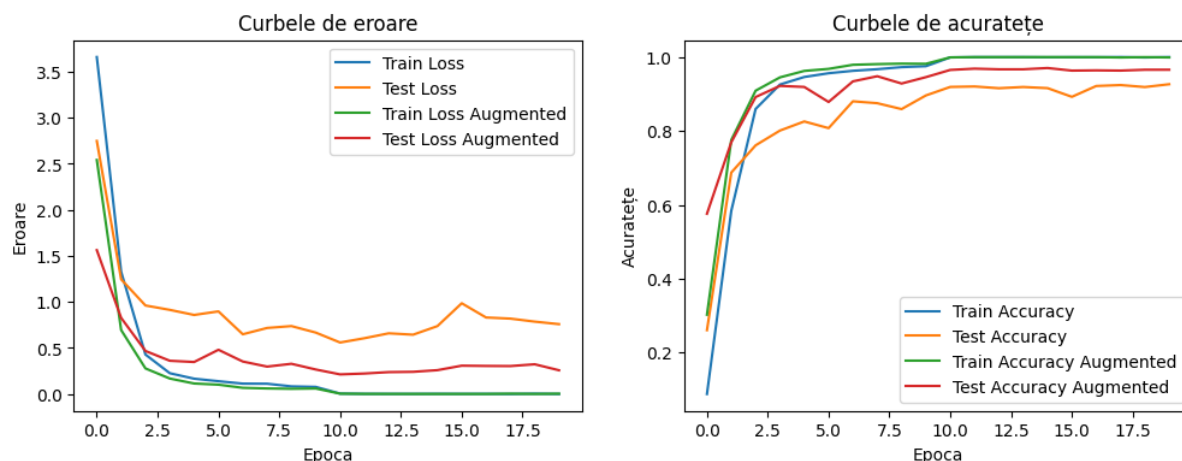
#### Fashion



Adaugarea augmentarii datelor imbunatateste performanta pe setul de test, cu o eroare mai mica si o acuratete mai mare comparativ cu modelul fara augmentari. Acuratetea pe setul de test ajunge aproape de cea de train, ceea ce indica o mai buna generalizare si reducerea overfitting-ului. Modelul cu augmentari pare mult mai bun.

#### Fruits360

Am folosit aici un learning rate de 0.001 pentru ambele rulari si am dat resize, am rulat pe 20 de epoci



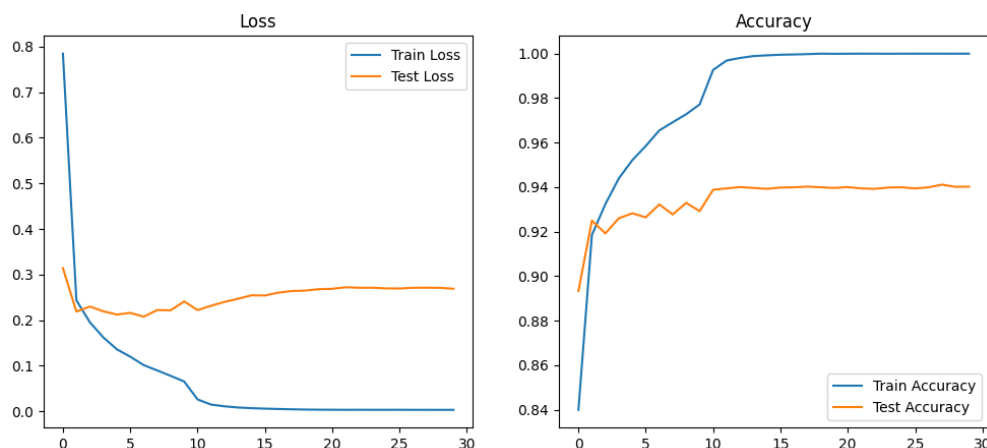
Graficul arata ca augmentarea datelor reduce eroarea pe setul de test si imbunatateste acuratetea, indicand o mai buna generalizare a modelului. Curbele pentru modelul augmentat sunt mai apropiate intre train si test, ceea ce arata ca augmentarea ajuta la prevenirea overfitting-ului. Modelul fara augmentare prezinta o diferenta mai mare intre train si test.

### 3.4 Utilizarea unei proceduri de finetuning peste arhitectura ResNet-18

Pentru punctul 3.4, am folosit ResNet-18 pre-antrenat pe CIFAR-10 pe seturile de date Fashion-MNIST si Fruits-360.

#### Fashion

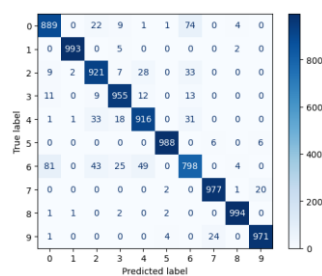
Pe durata antrenamentului, am utilizat imagini transformate in format RGB(le-am duplicat în 3 canale), am aplicat normalizarea si am folosit un scheduler pentru rata de invatare, care o reduce la fiecare 10 epoci. Parametrii folositi au fost o rata de invatare de 0.001 si un optimizar SGD cu momentum 0.9. Procesul a durat aproximativ 36 de minute



In graficul de Loss se vede ca modelul invata bine pe setul de antrenare, iar pe setul de test se stabilizeaza, dar este mai mare decat cel de pe train, ceea ce arata un usor overfitting. In graficul de acuratete, pe train acuratetea ajunge la 100%, iar pe test se stabilizeaza in jur de 92-94%, ceea ce arata ca modelul generalizeaza destul de bine.

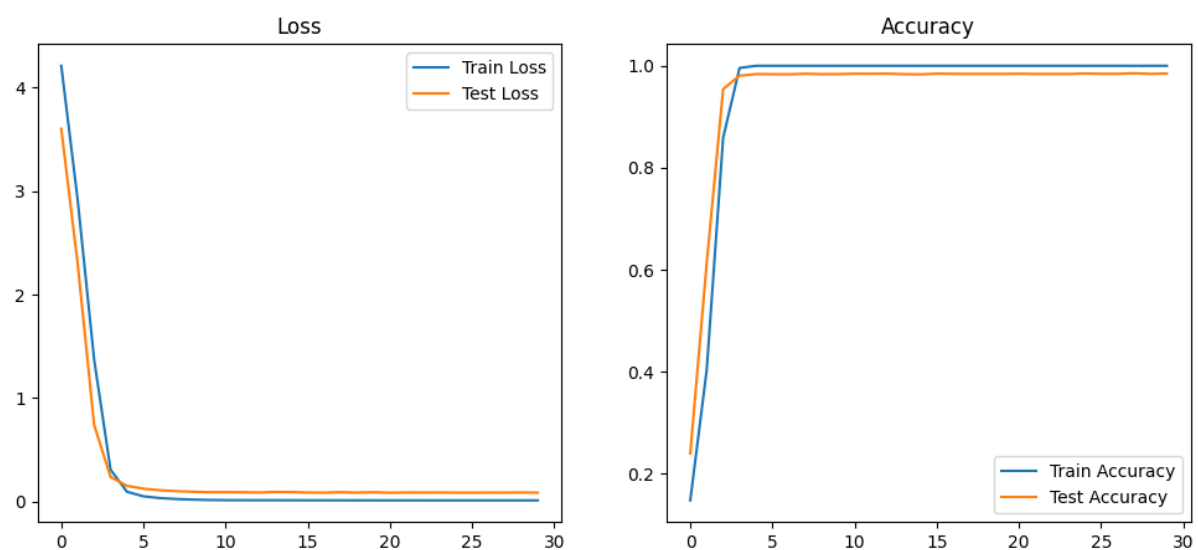
In concluzie, ResNet18 s-a descurcat bine pentru acest task, dar probabil se poate reduce overfitting-ul prin alte augmentari sau ajustari de hiperparametri.

Class	Precision	Recall	F1-Score	Support
0	0.90	0.89	0.89	1000
1	1.00	0.99	0.99	1000
2	0.90	0.92	0.91	1000
3	0.94	0.95	0.95	1000
4	0.91	0.92	0.91	1000
5	0.99	0.99	0.99	1000
6	0.84	0.80	0.82	1000
7	0.97	0.98	0.97	1000
8	0.99	0.99	0.99	1000
9	0.97	0.97	0.97	1000
accuracy			0.94	10000
macro avg	0.94	0.94	0.94	10000
weighted avg	0.94	0.94	0.94	10000



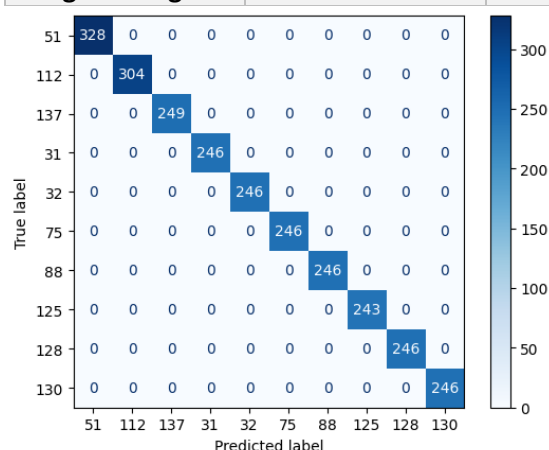
## Fruits360

Am aplicat o transformare de redimensionare pe setul de date, care este deja in format RGB. De asemenea, am utilizat normalizarea si un scheduler pentru rata de invatare, care scade la fiecare 10 epoci. Parametrii folositi au fost o rata de invatare de 0.001 si un momentum de 0.9. Procesul a durat aproximativ 56 de minute.



Modelul arata performanta excelenta, cu eroarea aproape de zero si acuratetea aproape de 100% atat pe train, cat si pe test. Lipsa unei diferente semnificative intre cele doua seturi sugereaza o generalizare foarte buna, iar modelul pare bine calibrat fara semne de overfitting. Aceasta ar putea fi o situatie ideala, dar ar trebui verificata autenticitatea datelor si procesul de antrenare.

Class	Precision	Recall	F1-Score	Support
<b>accuracy</b>			0.98	23619
<b>macro avg</b>	0.99	0.98	0.98	23619
<b>weighted avg</b>	0.99	0.98	0.98	



## Concluzie

MLP-ul pe attribute nu a performat foarte bine, avand rezultate mai slabe decat metodele din prima etapa, ceea ce arata ca acest tip de abordare nu este foarte eficient pentru aceste seturi de date.

MLP-ul aplicat direct pe imagini a avut rezultate similare cu cele mai eficiente metode din etapa anterioara, cum ar fi XGBoost. Totusi, pe setul Fashion-MNIST, SVM-ul a avut o performanta mai buna, cu o acuratete de 0.89 fata de 0.86 obtinuta de MLP.

In cazul retelei convolutionale, adaugarea augmentarilor a adus o imbunatatire semnificativa a performantei, facand ca modelul sa se apropie de rezultatele lui XGBoost (0.86 fata de 0.88 pe Fashion-MNIST). Acest lucru arata ca augmentarile de date sunt foarte utile pentru cresterea generalizarii modelului.

Reteaua preantrenata a fost cea mai performanta dintre toate metodele testate, reusind sa obtina o acuratete aproape de 1 atat pe setul Fashion-MNIST, cat si pe Fruits360. Acest rezultat evidentiaza avantajul utilizarii unui model preantrenat pentru sarcini complexe de clasificare a imaginilor.

## Hardware

Am rulat tema pe pc-ul personal cu 1080ti, 17 7700 si 16gb de ram.