

Tema 2 Învățare Automată

Clasificare Imagini FashionMNIST, Fruits-360

Alexandru Licuriceanu
alicuriceanu@stud.acs.upb.ro

1 Cerința 3.1 - MLP pe attributele extrase în etapa 1

În etapa 1 am extras attributele folosind metoda Histogram of Oriented Gradients (HOG) în combinație cu Principal Component Analysis (PCA) pentru reducerea dimensionalității, apoi am folosit SelectPercentile pentru a selecta un număr mai mic de attribute dintre cele rămase în așa fel încât:

- Pentru setul de date FashionMNIST să rămân cu 64 de attribute.
- Pentru setul de date Fruits-360 să rămân cu 128 de attribute.

1.1 FashionMNIST

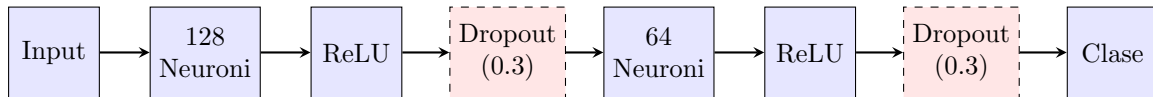


Table 1: Caracteristicile procesului de antrenare

Caracteristică	Valoare
Numărul de epoci de antrenare	50
Metode de regularizare	Straturi de Dropout
Funcția de eroare	Cross-Entropy Loss
Dimensiunea batch-ului	64
Optimizatorul folosit	Adam, Learning Rate = 0.001

1.2 Fruits-360

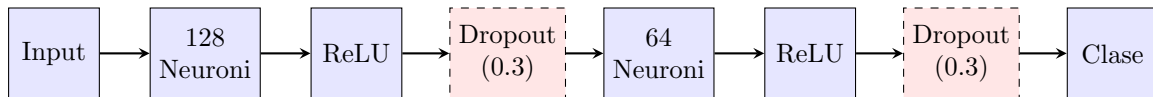


Table 2: Caracteristicile procesului de antrenare

Caracteristică	Valoare
Numărul de epoci de antrenare	30
Metode de regularizare	Straturi de Dropout, Weight Decay = 1e-5
Funcția de eroare	Cross-Entropy Loss
Dimensiunea batch-ului	64
Optimizatorul folosit	Adam, Learning Rate = 0.001

1.3 Curbe de eroare și acuratețe

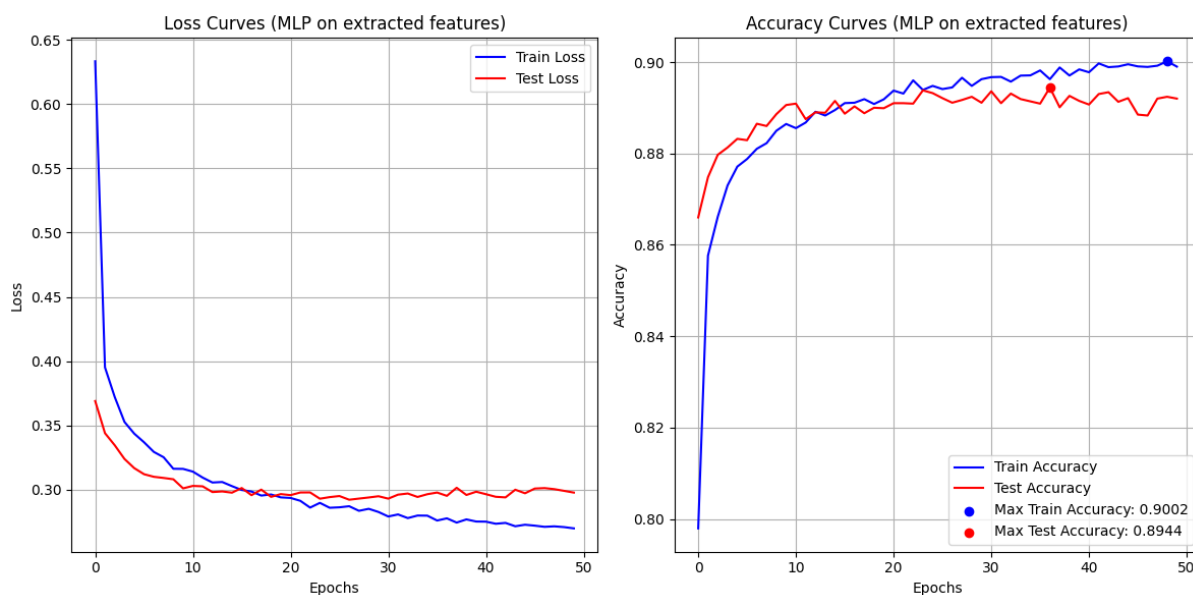


Figure 1: Curbele de antrenare pentru FashionMNIST

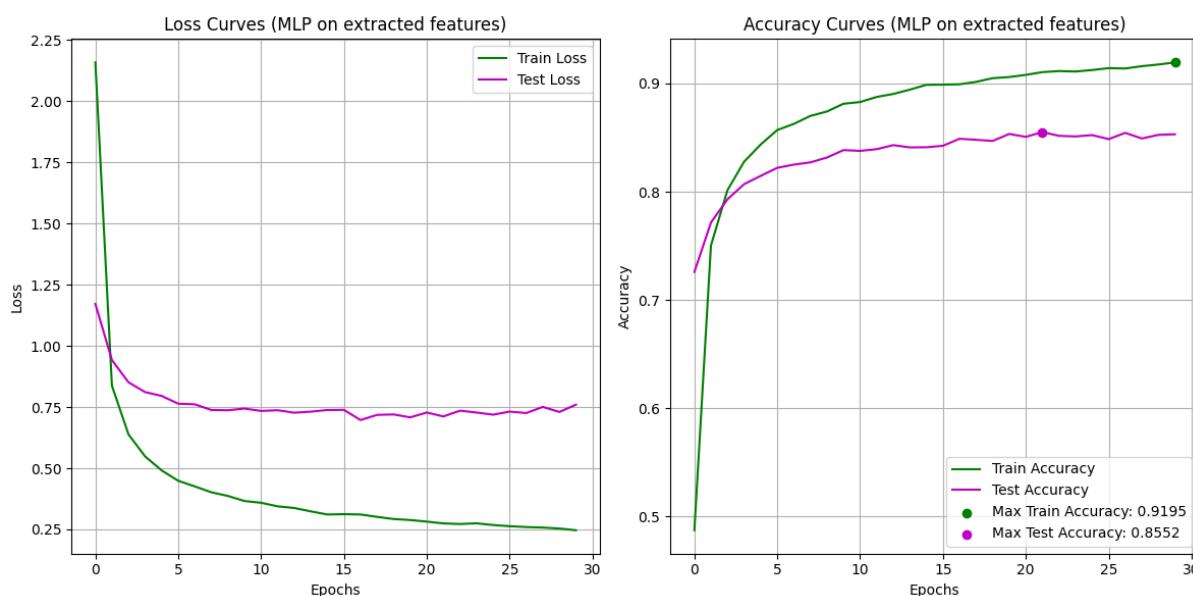


Figure 2: Curbele de antrenare pentru Fruits-360

Comparativ cu prima etapă, MLP-ul peste atributele extrase obține rezultate aproximativ la fel de bune precum algoritmi de IA clasici:

- FashionMNIST - Acuratețea maximă obținută de MLP pe setul de testare este de 0.89, iar cea mai bună acuratețe din etapa 1 este de 0.90 (SVM).
- Fruits-360 - Acuratețea maximă obținută de MLP este de 0.85, iar cea mai bună acuratețe din etapa 1 este de 0.88 (SVM).
- De menționat este că numărul de atribute diferă drastic între cele două etape, dar MLP-ul totuși reușește să obțină rezultate similare cu cele din etapa 1. De asemenea, probabil din această cauză modelul folosit pentru setul de date Fruits-360 nu reușește să mai învețe nimic nou începând cu

epoca 8-9 (PCA-ul a comprimat prea multe atribute). Curbele de eroare pentru Fruits indică o varianță crescută între loss-ul pe train și cel pe test, ceea ce pare a fi overfitting.

2 Cerința 3.2 - MLP direct peste imagini

Pentru o comparație cât se poate de obiectivă și corectă, am păstrat arhitectura MLP-ului utilizat la cerința anterioară pentru FashionMNIST și ușor modificată pentru Fruits-360. De asemenea, imaginile din Fruits-360 au fost scalate de la dimensiunea de 100x100 la 32x32 (prin testare am descoperit că diferența de performanță a modelului este practic inexistentă între cele două dimensiuni).

2.1 FashionMNIST

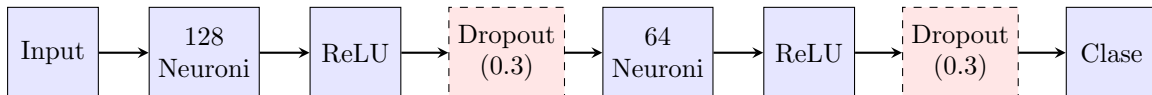


Table 3: Caracteristicile procesului de antrenare

Caracteristică	Valoare
Numărul de epoci de antrenare	30
Metode de regularizare	Straturi de Dropout
Funcția de eroare	Cross-Entropy Loss
Dimensiunea batch-ului	64
Optimizatorul folosit	Adam, Learning Rate = 0.001

2.2 Fruits-360

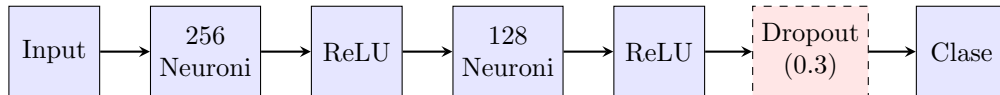


Table 4: Caracteristicile procesului de antrenare

Caracteristică	Valoare
Numărul de epoci de antrenare	30
Metode de regularizare	Strat de Dropout
Funcția de eroare	Cross-Entropy Loss
Dimensiunea batch-ului	64
Optimizatorul folosit	Adam, Learning Rate = 0.0005

2.3 Curbe de eroare și acuratețe

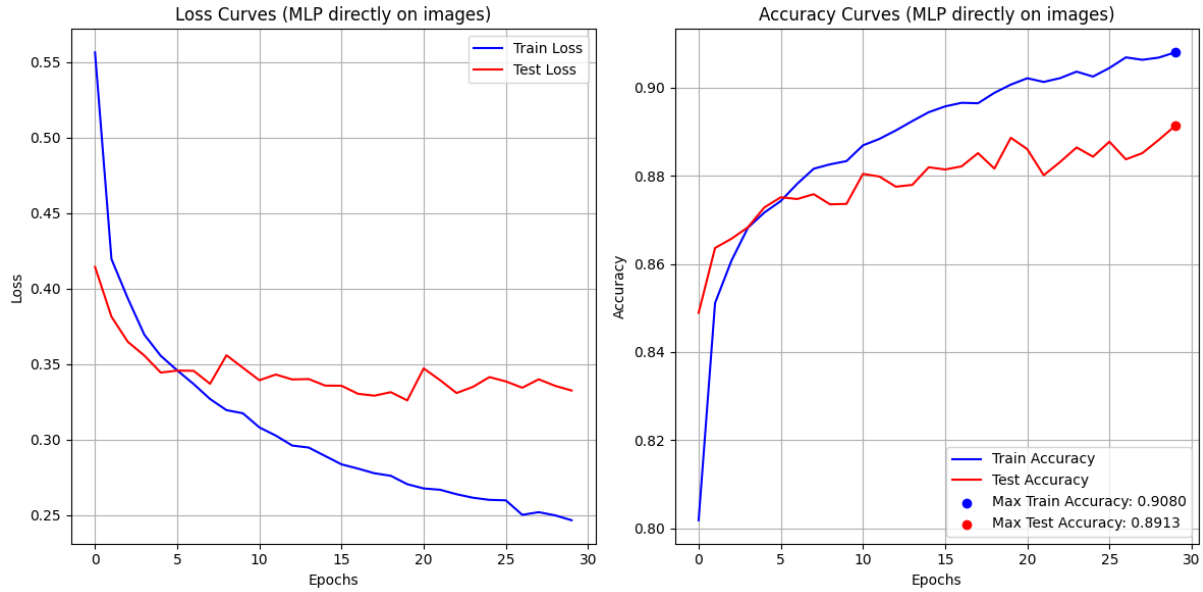


Figure 3: Curbele de antrenare pentru FashionMNIST

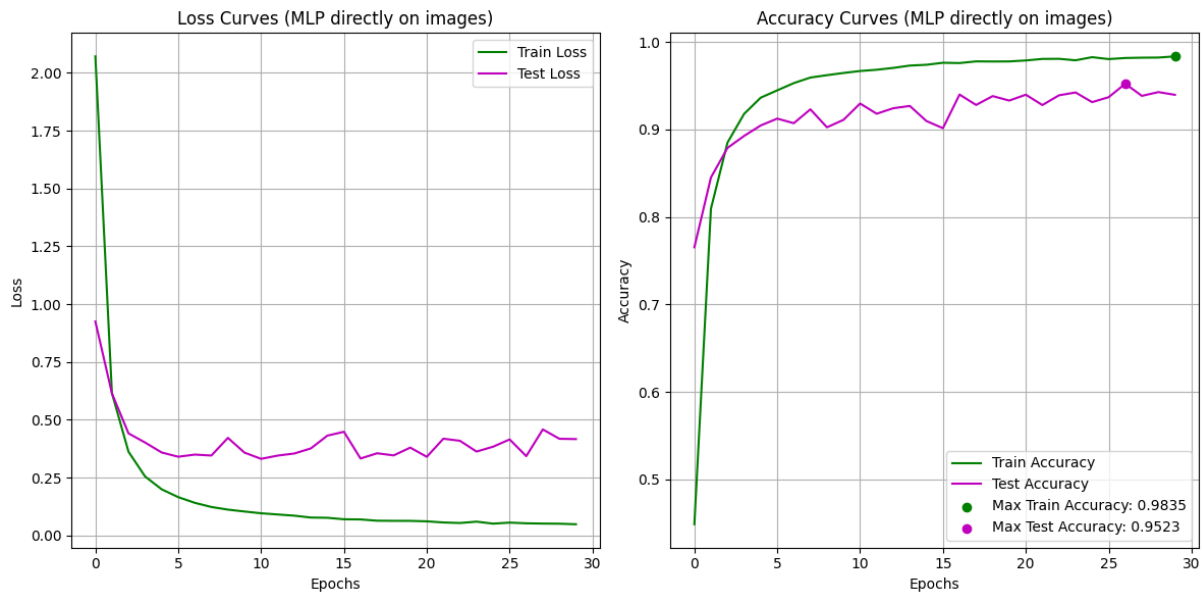
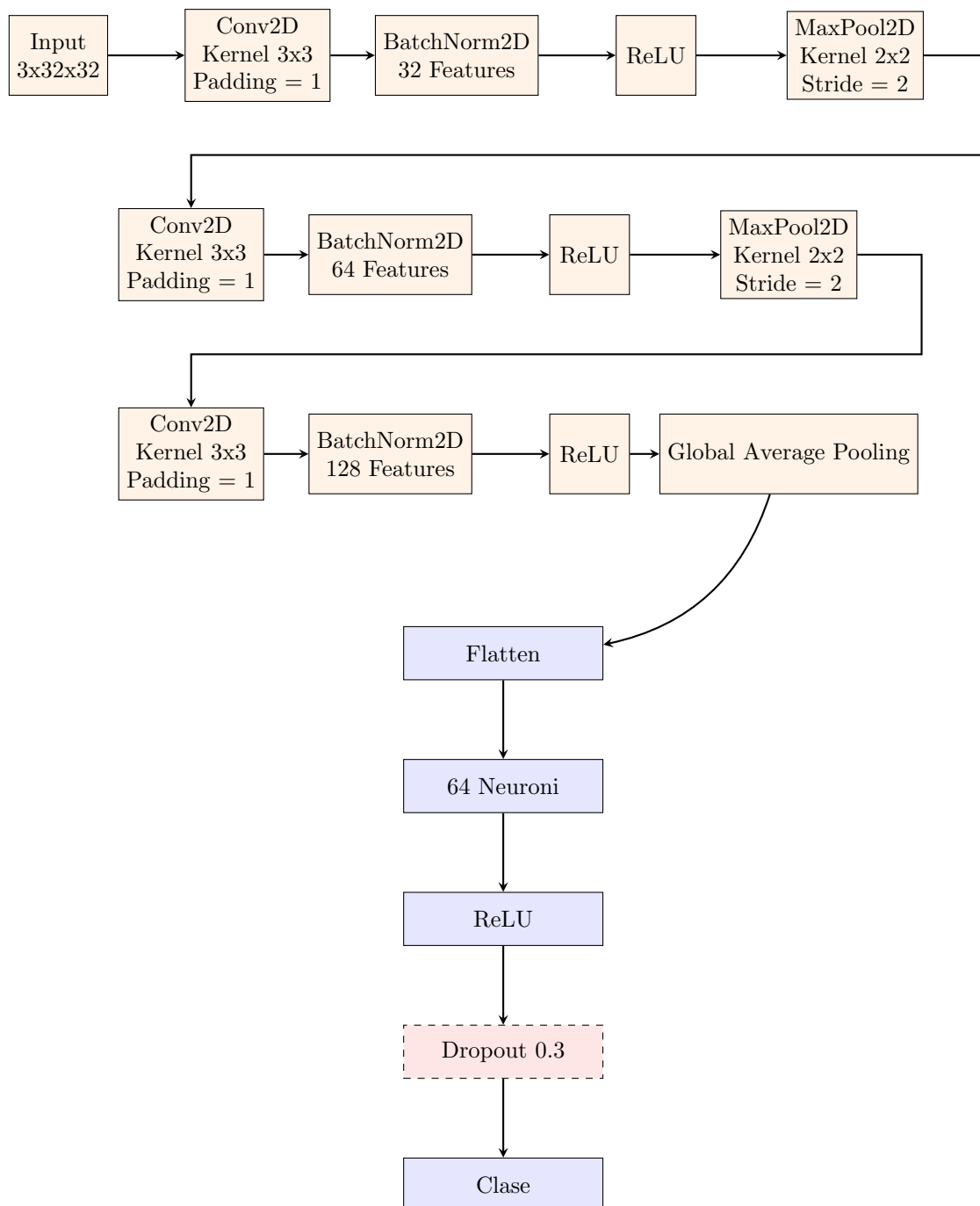


Figure 4: Curbele de antrenare pentru Fruits-360

Comparând cu MLP-ul care procesează atribute extrase, performanța MLP-ului aplicat direct peste imagini este aproape identică pentru setul de date FashionMNIST. Pentru arhitectura ușor modificată și folosită pe Fruits-360, se observă o creștere considerabilă în acuratețea modelului, ajungând la 0.95 pe setul de testare. Din nou se poate observa un ușor overfitting pe ambele seturi de date, însă nu la fel de accentuat precum cel de la cerința anterioară, pe setul Fruits.

3 Arhitectura de tip convoluțional

Pentru această cerință, am construit o rețea convoluțională foarte similară cu DeepConvNet, cuprinzând următoarele tipuri de straturi: Convoluțional, Global Pooling și straturile Fully-Connected. Am folosit aceeași rețea pentru ambele seturi de date, cât și pentru seturile de date augmentate.



Pentru ambele seturi de date, am folosit separat și augmentări în încercarea de a spori performanța modelului. Acestea includ: Resize la 32x32, RandomHorizontalFlip, RandomRotation cu maxim 10 grade și RandomResizedCrop cu dimensiunea de 32x32 și scale (0.8, 1.0).

Table 5: Caracteristicile procesului de antrenare - FashionMNIST

Caracteristică	Valoare
Numărul de epoci de antrenare	30
Metode de regularizare	Batch Normalization, Strat de Dropout
Funcția de eroare	Cross-Entropy Loss
Dimensiunea batch-ului	64
Optimizatorul folosit	Adam, Learning Rate = 0.0003 (0.001 pentru imaginile augmentate)

Table 6: Caracteristicile procesului de antrenare - Fruits-360

Caracteristică	Valoare
Numărul de epoci de antrenare	20
Metode de regularizare	Batch Normalization, Strat de Dropout
Funcția de eroare	Cross-Entropy Loss
Dimensiunea batch-ului	64
Optimizatorul folosit	Adam, Learning Rate = 0.0003 (0.001 pentru imaginile augmentate)

3.1 Curbe de eroare și acuratețe

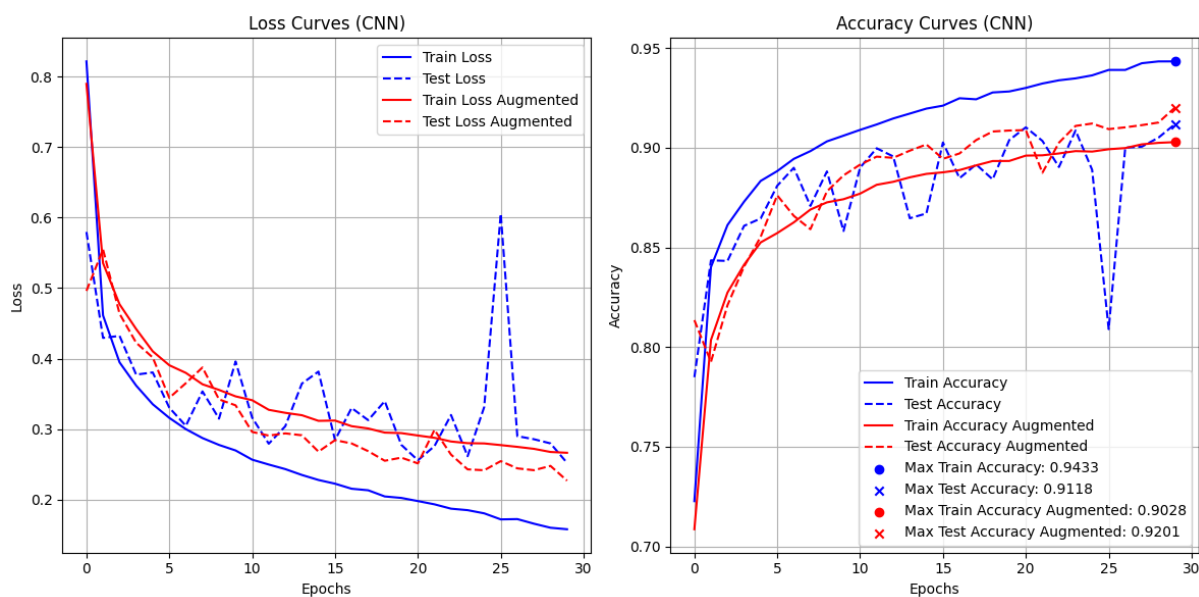


Figure 5: Curbele de antrenare pentru FashionMNIST

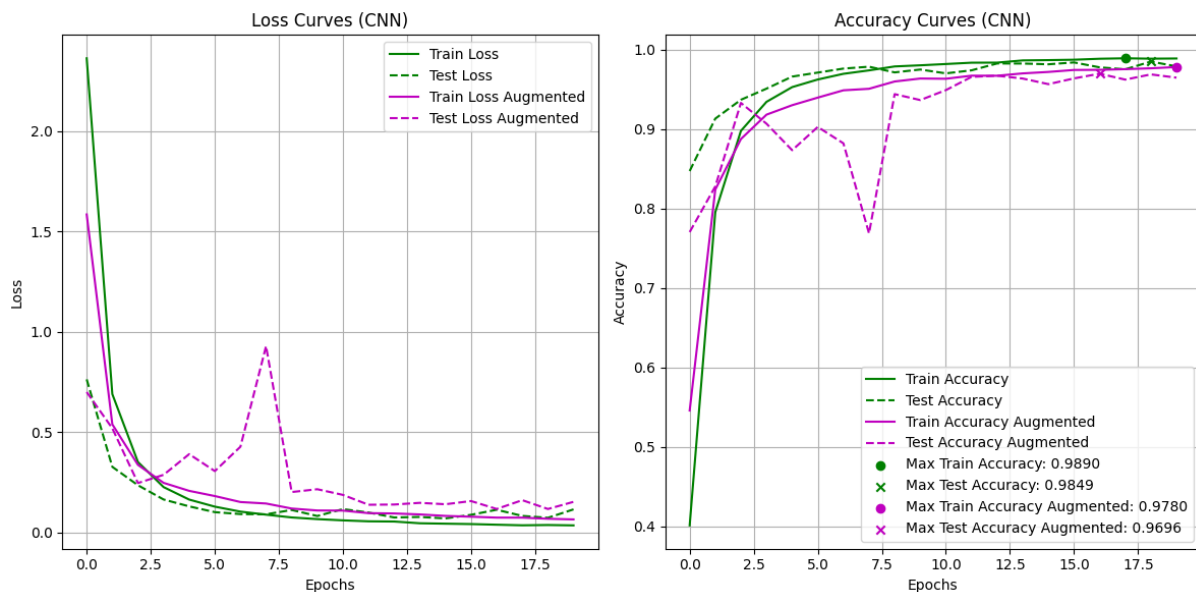


Figure 6: Curbele de antrenare pentru Fruits-360

De pe graficele prezentate, se poate observa că augmentările aduse nu produc o creștere semnificativă în performanța modelului, iar metricile pe seturile de date cu augmentări sunt la fel de bune sau puțin mai slabe decât cele pe seturile de date neaugmentate. O posibilă explicație ar putea fi că setul de test nu conține imagini cu variațiunile care rezultă din urma augmentărilor, iar astfel modelul învață imaginile augmentate, dar nu performează bine pe cele din setul de test (care nu sunt rotite, decupate, etc).

Cu toate acestea, rețeaua convoluțională are scoruri foarte bune, obținând acuratețea de 0.91 pe FashionMNIST neaugmentat, 0.92 augmentat iar 0.98 pe setul original Fruits-360 și 0.96 pe cel augmentat. Scorurile mult mai bune pentru setul de date Fruits-360, față de Fashion, se datorează cel mai probabil faptului ca fructele au caracteristici mai diferite între ele față de obiectele vestimentare (cum ar fi un tricou și o cămașă) și de asemenea 3 canale de culoare de unde se pot extrage informații comparat cu un singur canal de gri pentru Fashion.

Pentru Fashion modelul face puțin overfitting și pe setul neaugmentat, iar ceva mai puțin pe cel augmentat. Pentru setul de date Fruits modelul generalizează datele foarte bine atât pe imaginile augmentate, cât și pe cele neaugmentate.

4 Utilizarea unei proceduri de fine-tuning peste arhitectura ResNet

Pentru această cerință, am folosit un model ResNet-18, preantrenat pe ImageNet și modificat pentru a funcționa cu imaginile de rezoluție mică din CIFAR10. Pentru Fashion și Fruits am aplicat Resize pentru a le aduce la rezoluția de 32x32, și am normalizat imaginile astfel încât să aibă media și deviația standard a imaginilor din ImageNet.

În final, am înlocuit straturile Fully-Connected din ResNet cu un singur strat liniar care are ca intrare numărul de atribute și ca ieșire etichetele imaginilor. Parametrii de antrenare sunt aceiași pentru ambele seturi de date, după cum urmează:

Table 7: Caracteristicile procesului de antrenare - FashionMNIST, Fruits-360

Caracteristică	Valoare
Numărul de epoci de antrenare	10
Metode de regularizare	Batch Normalization (din ResNet)
Funcția de eroare	Cross-Entropy Loss
Dimensiunea batch-ului	64
Optimizatorul folosit	SGD, Learning Rate = 0.0005, Momentum = 0.9
Scheduler	StepLR, Step Size = 3, Gamma = 0.1

4.1 Curbe de eroare și acuratețe

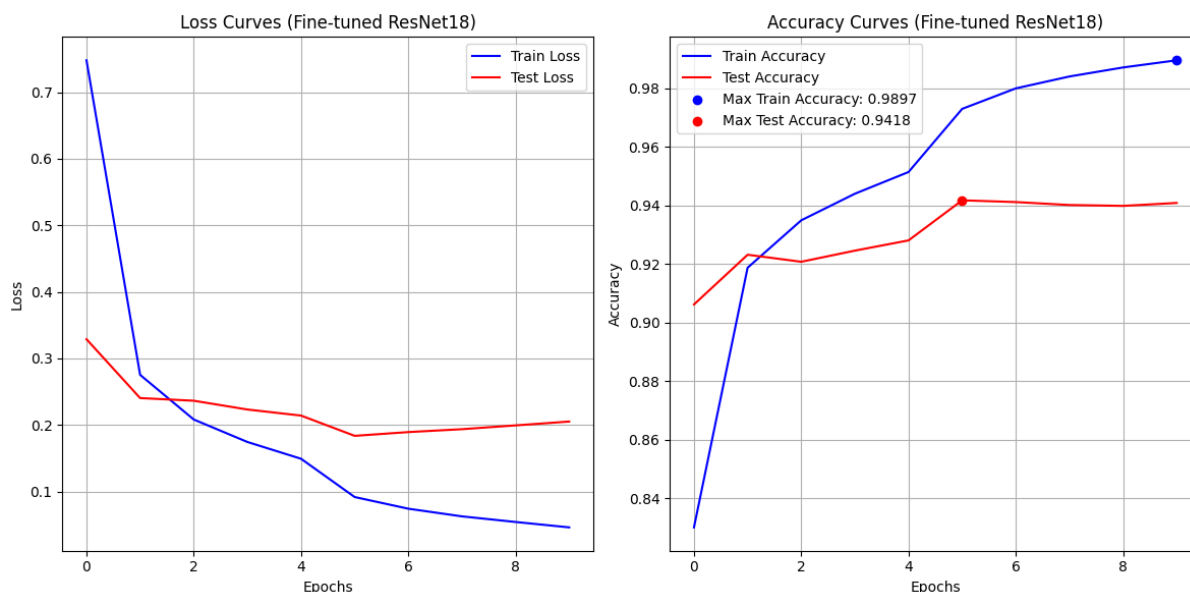


Figure 7: Curbele de antrenare pentru FashionMNIST

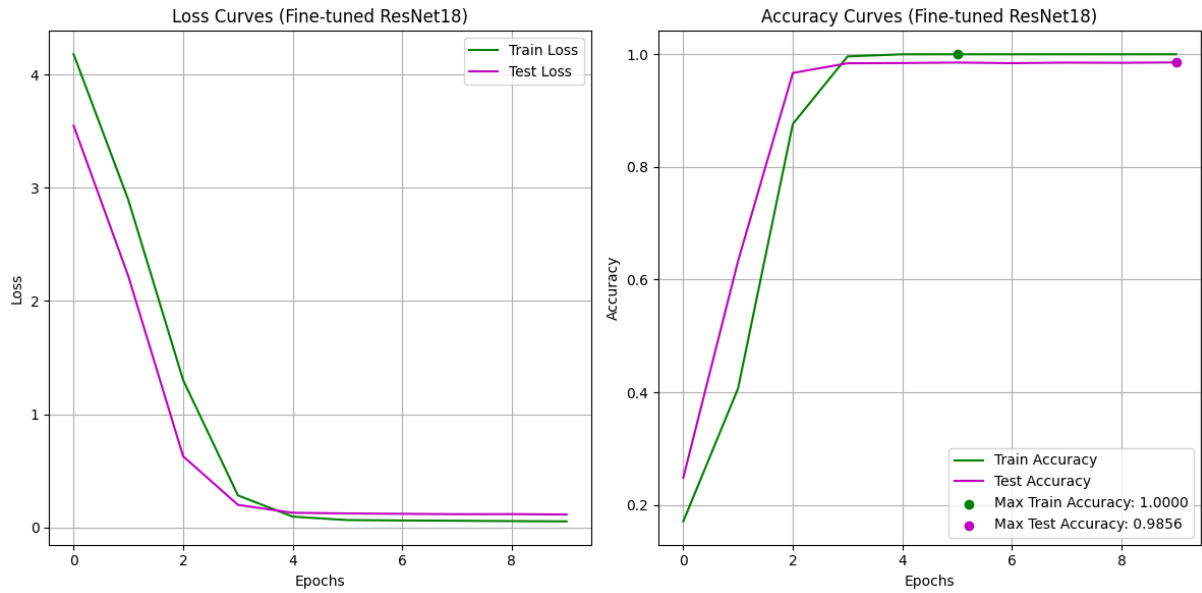


Figure 8: Curbele de antrenare pentru Fruits-360

Pe setul de date Fashion rețeaua obține rezultate similare cu cele anterioare, 0.94 acuratețe pe setul de test, însă în mult mai puține epoci. Pe de altă parte, pe setul de date Fruits ResNet preantrenat se comporta aproape perfect, obține acuratețe maximă pe setul de antrenare, și în același timp cea mai mare acuratețe pe setul de testare, 0.98.

5 Evaluare comparativă

Table 8: Comparație Arhitecturi - FashionMNIST

Arhitectură	Acuratețe Antrenare	Acuratețe Testare	Timp	Epoci
MLP peste attribute	0.90	0.89	3m 20s (CPU)	50
MLP peste imagini	0.90	0.89	2m 51s (CPU)	30
CNN (fără augmentări)	0.94	0.91	7m 30s (GPU)	30
CNN (cu augmentări)	0.90	0.92	18m 22s (GPU)	30
ResNet-18 Fine-Tuning	0.98	0.94	8m 20s (GPU)	10

Table 9: Comparație Arhitecturi - Fruits-360

Arhitectură	Acuratețe Antrenare	Acuratețe Testare	Timp	Epoci
MLP peste attribute	0.91	0.85	2m 26s (GPU)	30
MLP peste imagini	0.98	0.95	26m 30s (GPU)	30
CNN (fără augmentări)	0.98	0.98	22m 20s (GPU)	20
CNN (cu augmentări)	0.97	0.96	31m 20s (GPU)	20
ResNet-18 Fine-Tuning	1.00	0.98	18m 10s (GPU)	10

Diferențele între metode sunt explicate prin complexitatea arhitecturii și utilizarea augmentărilor. Arhitecturile simple cum sunt MLP-urile sunt rapide, dar au acuratețe scăzută, în timp ce arhitecturile mai complexe (CNN, ResNet-18) oferă performanță ridicată, însă necesită mai mult timp pentru a fi antrenate.

Varianta cu modelul ResNet-18 este liderul evident pentru ambele seturi de date, datorită fine-tunării pe arhitectura pre-antrenată, ceea ce îi permite să extragă caracteristici mai bune.

Comparând cu etapa 1, rezultatele sunt cel puțin la fel de bune pe ambele seturi de date și pentru toate abordările, dar CNN-urile și în special ResNet fine-tunat obțin scoruri semnificativ mai bune decât algoritmi clasici de inteligență artificială din etapa 1 care abia reușeau să treacă de acuratețe 0.90 pe testare.

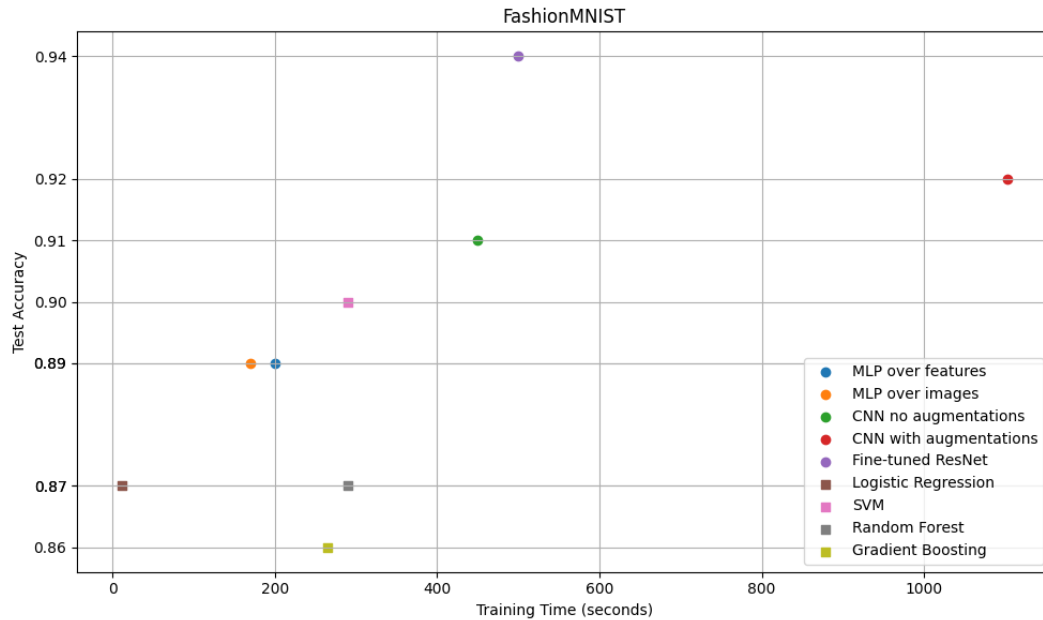


Figure 9: Comparație cu etapa 1 - FashionMNIST

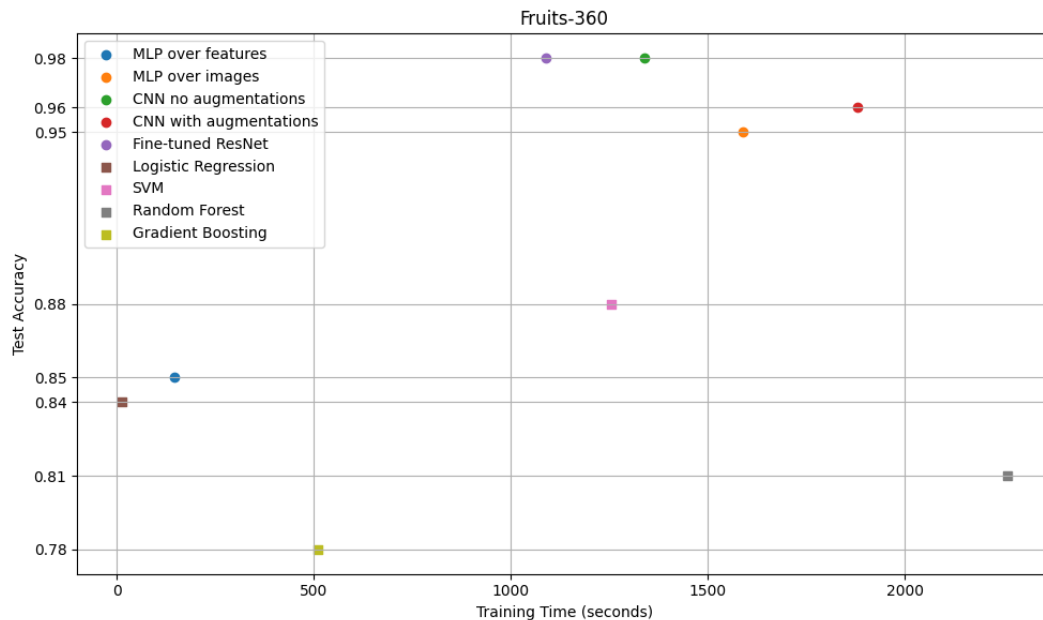


Figure 10: Comparație cu etapa 1 - Fruits-360